

<http://v3.espacenet.com/publicationDetails/biblio?adjacent=true&KC=A&date=19990...> 2009/05/01

【図 31】

-----  
[ 文書管理作業 ]

+名前	第 1 章管理作業
+対象文章	「第1章」
+進捗	“ 終了 ”
+進捗報告	“ 文章第 1 章の進捗は、終了で 10 行です ”

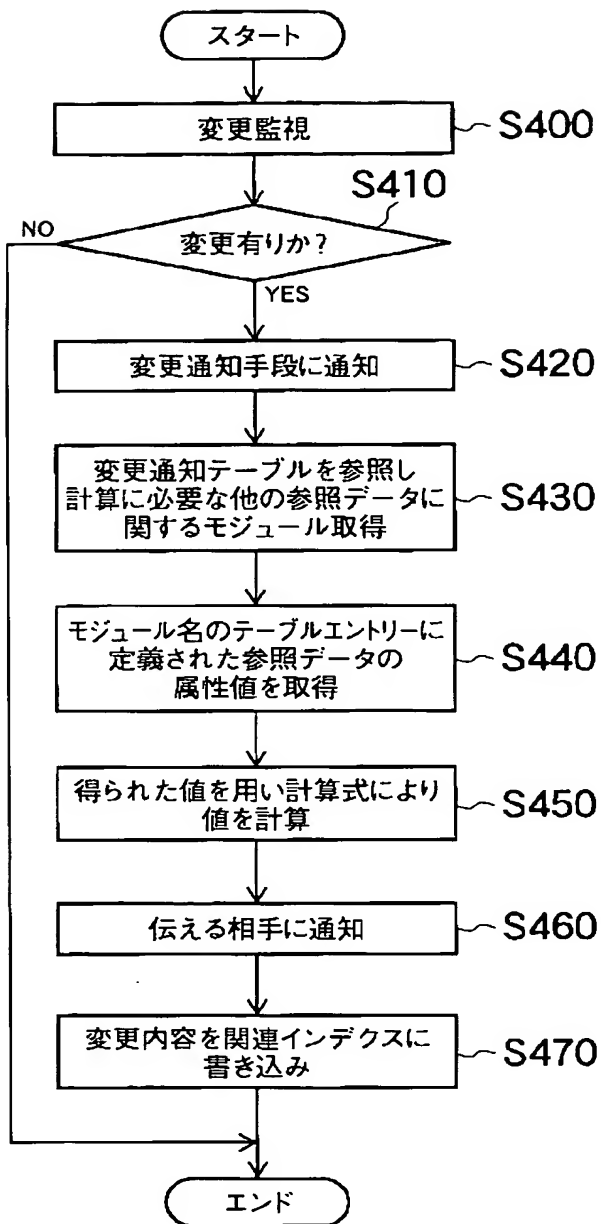
----------  
[ 全体管理作業 ]

+名前	“ 報告書作成管理 ”	
+状態	作業中	
+進捗	名前	状態
	“ 第 1 章 ”	終了
	“ 第 2 章 ”	作業中
	“ まとめ ”	作業中
+要約	はじめに、問題点を指摘した。解決方法を提示して実験をおこなった。評価をおこなった。	
+関連文書	{ 「 第 1 章 」, 「 第 2 章 」, 「 まとめ 」 }	
○管理者	D	

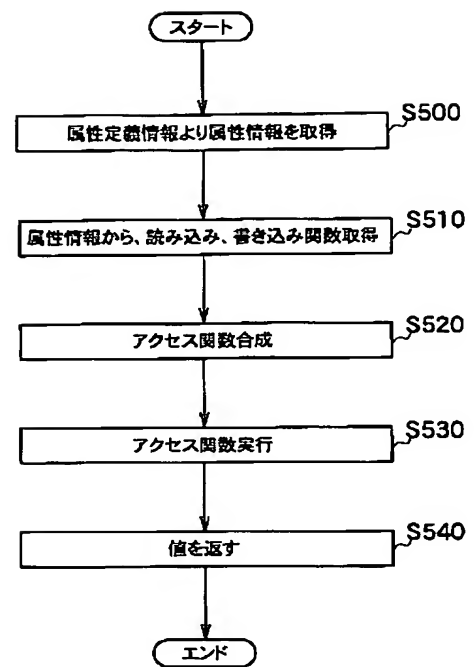
-----



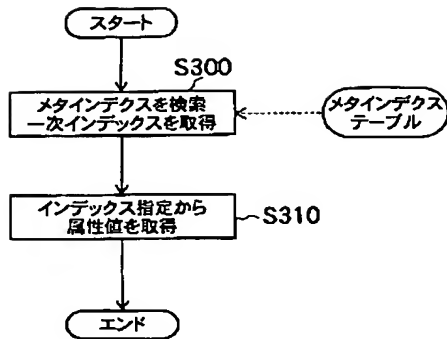
【図 27】



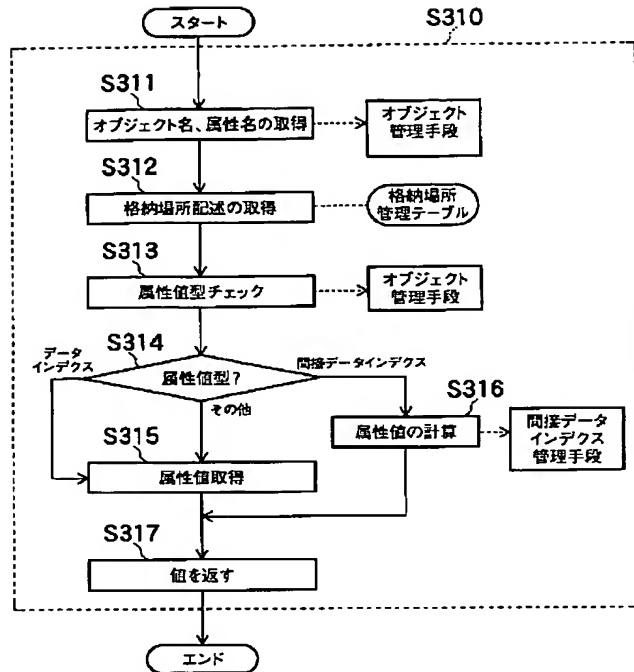
【図 32】



【図22】



【図23】



【図26】

格納場所定義,	属性値
「第1章」#タイトル,	第1章
「第1章」#データ,	section1.obj
「第1章」#名前,	第1章
「第1章」#内容,	"はじめに、この文章は...である。"
「第1章」#行数,	10;
「第1章」#要約,	"はじめに"
「第1章」#作成者名,	A
「第1章」#コメント,	NULL
「第1章」#作成者,	A
「第1章」#親文書,	「報告書」

【図30】

1. 全ての【文章】オブジェクトに対して【文章管理作業】オブジェクトXを対応させ、  
属性 = "【文章管理作業】オブジェクトX.対象文章"に元のオブジェクトのリファレンスを組込む。
2. 全ての文章管理と1つの【全体管理作業】オブジェクトYを対応させ、  
属性 = "【全体管理作業】オブジェクトY.関連作業"に元のオブジェクトのリファレンスを組込む。

【図 18】

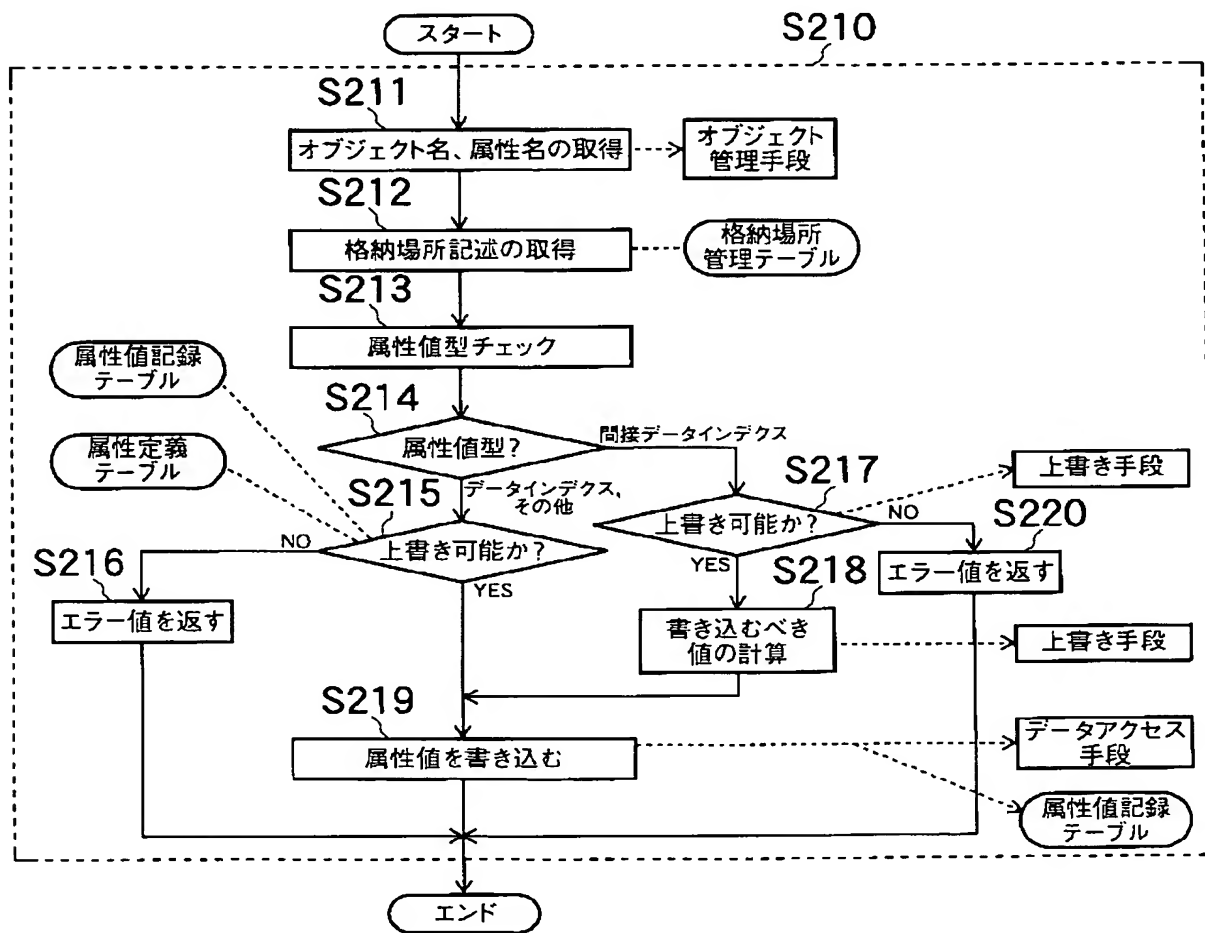
格納場所定義	値
「報告書作成管理」 #進捗#1#1	「第 1 章」
「報告書作成管理」 #進捗#1#2	作業中
「報告書作成管理」 #進捗#2#1	「第 2 章」
「報告書作成管理」 #進捗#2#2	初期状態

【図 19】

オブジェクト名 = 「報告書作成管理」

「報告書作成管理」. 進捗. 第 1 章. 名前	「報告書作成管理」 #進捗#1#1
「報告書作成管理」. 進捗. 第 1 章. 状態	「報告書作成管理」 #進捗#1#2
「報告書作成管理」. 進捗. 第 2 章. 名前	「報告書作成管理」 #進捗#2#1
「報告書作成管理」. 進捗. 第 2 章. 状態	「報告書作成管理」 #進捗#2#2

【図 21】



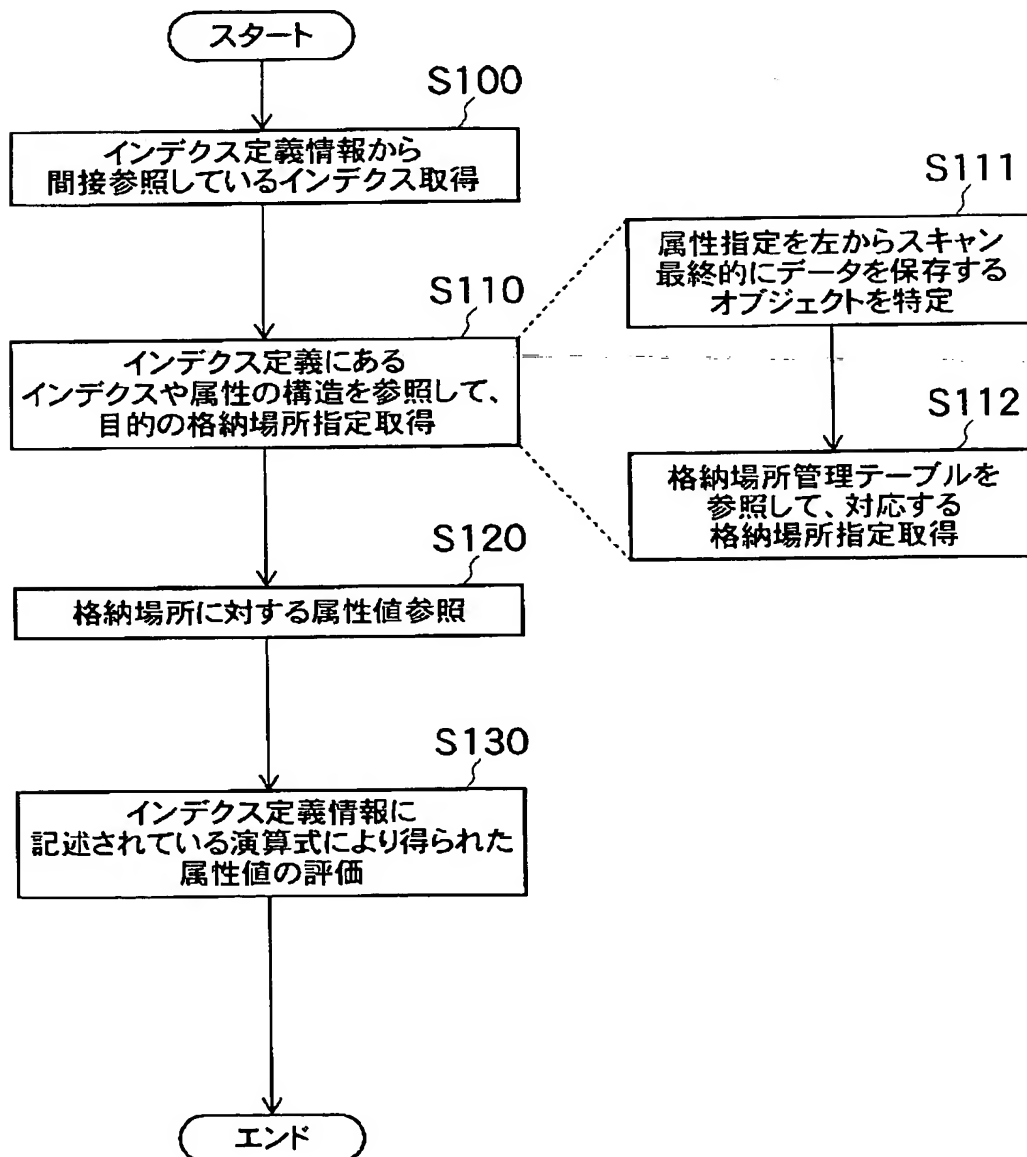
【図 25】

データタイプ	アクセスタイプ	関数名	関数ポインタ
doc,	read,	youyaku,	*youyaku
doc,	write,	NA,	—
doc,	read,	GetNaiyo,	*GetNaiyo
doc,	write,	PutNaiyo,	*PutNaiyo

【図 1 5】

メタインデクス、	実際のインデクス
“ 重要、要約、結論、文書 ”	まとめ、要約
進歩に関連する行数	第 1 章、行数
進歩に関連する予定行数は	第 1 章作成、目標行数

【図 1 6】



【図 8】

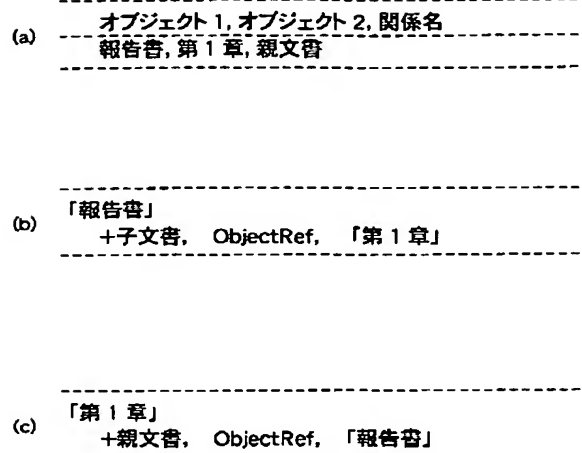
```

objectCreate([書類],「報告書」)
objectCreate([文章],「第1章」)
objectCreate([文章],「第2章」)
objectCreate([文章],「まとめ」)
objectCreate([管理作業],「報告書作成管理」)
objectCreate([人],「A氏」)
objectCreate([人],「B氏」)
objectCreate([人],「C氏」)
objectCreate([人],「D氏」)
objectCreate([作業],「報告書作成」)

relationAssert(「報告書」,「第1章」,親文章)
relationAssert(「報告書」,「第2章」,親文章)
relationAssert(「報告書」,「まとめ」,親文章)
relationAssert(「第1章」,「A氏」,作成者)
relationAssert(「第2章」,「B氏」,作成者)
relationAssert(「まとめ」,「C氏」,作成者)
relationAssert(「報告書」,「D氏」,作成者)
relationAssert(「報告書作成」,「D氏」,作業者)
relationAssert(「報告書作成」,「報告書」,作業対象)
relationAssert(「報告書作成管理」,「報告書」,管理対象)

```

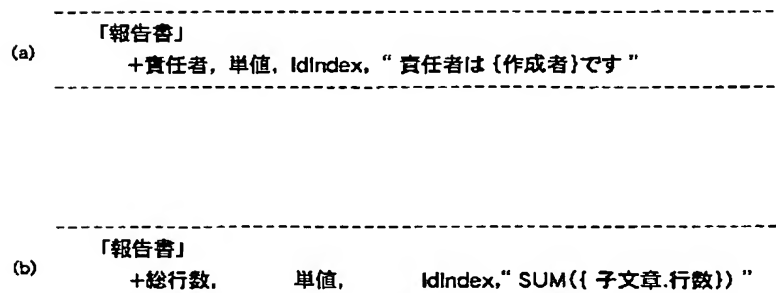
【図 1 1】



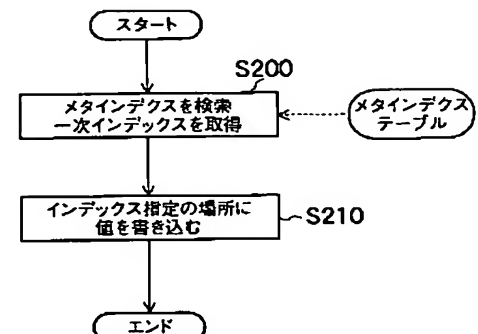
【図 1 3】

格納場所定義,	属性値
「第 1 章」 #タイトル,	第 1 章
「第 1 章」 #データ,	section1.obj
「第 1 章」 #名前,	第 1 章
「第 1 章」 #内容,	“はじめに、この文章は...である。”
「第 1 章」 #行数,	10;
「第 1 章」 #要約,	“はじめに”
「第 1 章」 #作成者名,	A
「第 1 章」 #コメント,	NULL
「第 1 章」 #作成者,	A
「第 1 章」 #親文章,	「報告書」

【図 1 4】



【図 2 0】





【図 6】

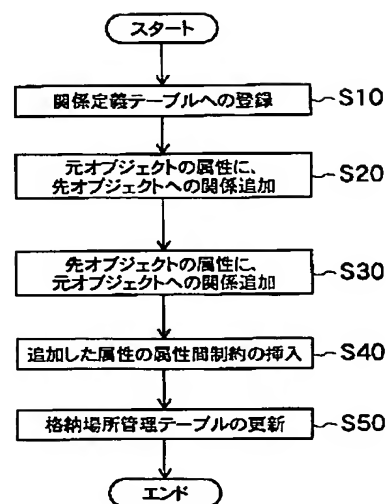
クラス定義	
属性名	属性型, 属性値型, 初期値, アクセス権
[万物]	
+名前	単値, STRING
+データ	単値, DataRef,,, rw
[文章] is_a 万物	
+タイトル	単値, IdIndex, "{名前}",
○作成者	
[章目] is_a 文章	
+内容	単値, DataIndex, "{doc, read, GetNaiyo, null}, {doc, write, PutNaiyo, null}", rw
+行数	単値, DataIndex, "{doc, read, line, VALUE}", r
+要約	単値, DataIndex, "{doc, read, youyaku, VALUE}", r
+作成者名	単値, IdIndex, "{作成者.名前}", rw
+コメント	単値, STRING, "", rw
○親文章	
[書類] is_a 文章	
+管理者名	単値, IdIndex, "{管理者.名前}"
+管理者	単値, IdIndex, "{作成者}"
○子文章	
[作業] is_a 万物	
+名前	単値, STRING,,
○作業対象	
○作業者	
[管理作業] is_a 作業	
+進歩	テーブル, IdIndex, "{関連文章.<名前=key,行数>}"
+関連文章	リスト, IdIndex, "{管理対象.子文章}"
○管理対象	
[人] is_a 万物	
+性別	単値, STRING

The diagram illustrates the system architecture with the following components and their interconnections:

- 外部アプリケーション** (External Application) connects to the **表示部** (Display Unit).
- 表示部** (800) contains **入出力手段** (Input/Output Means) and **表示手段** (Display Means). It is linked to **T810**.
- レイアウト定義テーブル** (Layout Definition Table) is connected to the **表示手段**.
- プラグイン部** (700) is a central component receiving input from the **表示部** and connecting to several other modules.
- クラス・同情報管理部** (610) and **新クラス管理手段** (620) are part of the plugin module's class management functions.
- オブジェクト管理部** (100) includes **クラス管理手段** (110), **インスタンス管理手段** (120), and **関係管理手段** (130).
- データ登録部** (200) handles data registration and is connected to the object management module.
- 属性定義テーブル** (T10), **クラス管理テーブル** (T60), **オブジェクト管理テーブル** (T70), **属性値記録テーブル** (T20), and **関係定義テーブル** (T80) are associated with the database layer.
- データベース** (DB) is the core storage, accessed by the **データアクセス部** (300) and **格納場所管理テーブル** (T90).
- メタインデックス管理手段** (440) manages meta-indexes, interacting with the **索引管理手段** (410) and the **メタインデックステーブル** (T40).
- 変更管理手段** (520) and **変更検知手段** (510) handle changes, linked to tables like **変更検知テーブル** (T110), **変更通知テーブル** (T120), **変更反映テーブル** (T130), and **変更反映手段** (530).
- 関連データインデックス管理手段** (420) and **上書き手段** (430) manage related data indexes and overwriting.
- アクセント関数テーブル** (T100) is used for access control functions.

【図 10】

	記号	意味
属性型	SingleValue	…単値
	List	…単値のリスト
	Table	…単値のテーブル
属性値型	STRING	文字列
	NUM	数字
	IdIndex	間接データインデクス情報
	ObjectRef	オブジェクトへのリファレンス情報
	DataRef	管理しているデータへのリファレンス情報
	DataIndex	データインデクス情報
		{データタイプ,アクセス型,アクセス関数,引数}



【図 12】

インデクス,	格納場所
報告書.子文章.「第 1 章」	報告書#子文章#1
報告書.子文章#1	報告書#子文章#1
報告書.子文章.( 名前 = 「第 1 章」 )	報告書#子文章#1
第 1 章.報告書名	報告書#名前

【图 24】

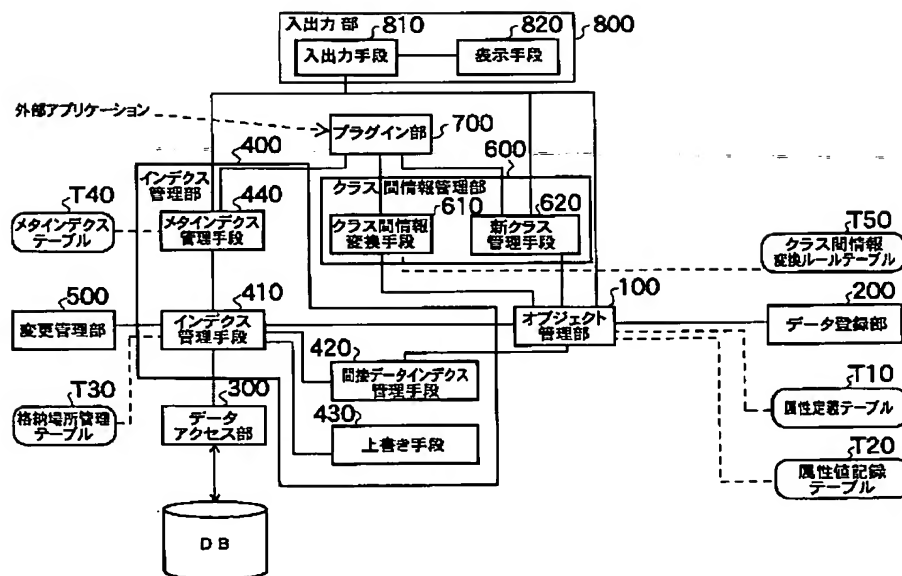
「報告書作成管理」 +進捗	テーブル, lcIndex, "(関連文章 (名前 = key, 行数))"	格納場所情報 「第1章」#コメント	属性値 "この章には大幅修正が必要"
------------------	--	----------------------	-----------------------

4 0 0 インデクス管理部  
 4 1 0 インデクス管理手段  
 4 2 0 間接データインデクス管理手段  
 4 3 0 上書き手段  
 4 4 0 メタインデクス管理手段  
 5 0 0 変更管理部  
 5 1 0 変更検知手段  
 5 2 0 変更通知手段  
 5 3 0 変更反映手段  
 6 0 0 クラス間情報管理部  
 6 1 0 クラス間情報変換手段  
 6 2 0 新クラス管理手段  
 7 0 0 プラグイン部  
 8 0 0 入出力部  
 8 1 0 入出力手段

\* 8 2 0 表示手段  
 T 1 0 属性定義テーブル  
 T 2 0 属性値記録テーブル  
 T 3 0 格納場所管理テーブル  
 T 4 0 メタインデクステーブル  
 T 5 0 クラス間情報変換ルールテーブル  
 T 6 0 クラス管理テーブル  
 T 7 0 オブジェクト管理テーブル  
 T 8 0 関係定義テーブル  
 10 T 9 0 データアクセス手順記録テーブル  
 T 1 0 0 アクセス関数テーブル  
 T 1 1 0 変更検知テーブル  
 T 1 2 0 変更通知テーブル  
 T 1 3 0 変更反映テーブル  
 \* T 1 4 0 レイアウト定義テーブル

【図 1】

【図 7】



【図 3】

【図 4】

関係名, 元クラス, 先クラス, 元ロール, 先ロール, 属性間制約  
 [編集作業], [人], [出版物], "編集対象", "編集者", なし

関係名, 元クラス, 先クラス, 元ロール, 先ロール, 属性間制約  
 [章目関係], [章目], [報告], "親文章", "子文章",  
 "[章目].報告書名<=[報告].名前"  
 [管理行為], [人], [管理], "管理する", "管理される"

【図 9】

dataRegister (「第1章」. データ, section1.doc);  
 dataRegister (「第2章」. データ, section2.doc);  
 dataRegister (「まとめ」. データ, matome.doc);  
 dataRegister (「報告書」. データ, report.doc);

【0226】さらに、既存のクラスと、新しく組み込まれたクラスとの間の対応関係を管理する機能を有するので、新たなクラスが新設された場合にも、これに対応したオブジェクトを再定義することなく、既存のオブジェクト体系の内容を変換利用することが可能となるという効果が得られる。これにより、部品データに関連するオブジェクトのメンテナンスを容易にすることができる。

【0227】また、外部アプリケーションからのビューを、既存のオブジェクトの適切なインデックスに変換する機能を有するので、外部からのビューの変更に容易に対処し、新規アプリケーションの組み込みやアプリケーションの変更への対応を図ることが可能となるという効果が得られる。これにより、さらに、データの再利用性、データの共有性を向上させることができる。

【0228】さらに、データの変更が生じた場合に、対応するオブジェクトとともに、関連する他のオブジェクトに、変更内容を随時反映する機能を有するので、データおよびこれに関連するオブジェクト間の一貫性を維持することが可能となるという効果が得られる。これにより、さらに、容易に保有する情報を管理、メンテナンスすることができる。

【0229】このように、本発明を用いれば、部品化されたデータを、外部の種々のアプリケーション、ユーザーから容易にアクセス、操作でき、また、保有するデータ、オブジェクト、外部アプリケーションが、変更、追加された際にも、そのデータの利用率を損なうことがないというデータ環境を提供することとなる。従って、システムの柔軟性を確保し、情報資産、ソフトウェア資産の十分かつ汎用的な活用を図ることができるのであり、産業上その効果のきわめて大きい発明である。

#### 【図面の簡単な説明】

【図1】本発明に係る情報管理装置の機能構成を示す機能ブロック図である。

【図2】本発明の実施形態に係る情報管理装置の機能構成を示す機能ブロック図である。

【図3】関係定義の例を示す図である。

【図4】関係定義テーブルの内容を示す図である。

【図5】本発明の実施形態における属性型と属性値型の内容を示す図である。

【図6】クラス定義テーブルの内容を示す図である。

【図7】属性型がテーブルであった場合の展開の例を示す図である。

【図8】オブジェクト管理部用初期化ファイルの内容を示す図である。

【図9】データ登録用関数定義を示す図である。

【図10】本発明の実施形態における関係挿入の手順を示すフローチャートである。

【図11】関係挿入において追加されるテーブルエンタリーを示す図である。

【図12】格納場所管理テーブルの内容を示す図であ

る。

【図13】属性値記録テーブルの内容の一部を示す図である。

【図14】間接データインデックス追加により追加される属性定義テーブルのエンタリーを示す図である。

【図15】メタインデックステーブルの内容を示す図である。

【図16】間接データインデックスによる属性値アクセスの処理手順を示すフローチャートである。

【図17】属性型がテーブルである場合の属性定義の例を示す図である。

【図18】属性値記録テーブルの内容の一部を示す図である。

【図19】格納場所管理テーブルの内容の一部を示す図である。

【図20】メタインデックス管理手段におけるメタインデックス記述を用いたデータ書き込みの処理手順を示すフローチャートである。

【図21】メタインデックス記述を用いたデータ書き込みの、インデックス管理手段内における処理手順を示すフローチャートである。

【図22】メタインデックス管理手段におけるメタインデックス記述を用いたデータ読み込みの処理手順を示すフローチャートである。

【図23】メタインデックス記述を用いたデータ読み込みの、インデックス管理手段内における処理手順を示すフローチャートである。

【図24】属性値記録テーブルの内容の一部を示す図である。

【図25】アクセス関数テーブルの内容を示す図である。

【図26】属性値記録テーブルの内容を示す図である。

【図27】変更管理部における処理手順を示すフローチャートである。

【図28】変更管理部が用いる各テーブルの定義内容を示す図である。

【図29】新クラス定義の例を示す図である。

【図30】クラス間情報変換ルールテーブルの定義内容を示す図である。

【図31】生成された仮想インスタンスの内容を示す図である。

【図32】データインデックスを用いたデータアクセスの処理手順を示すフローチャートである。

#### 【符号の説明】

100 オブジェクト管理部

110 クラス管理手段

120 インスタンス管理手段

130 関係管理手段

200 データ登録部

300 データアクセス部

で、戻り値の属性型が単値の場合には、そのままインデックス文字列を置き換え、またリストやテーブルの属性型の値が得られた場合には、レイアウト言語に合わせたリストやテーブルを生成し、これに対応した置き換え表示を行う。

【0212】尚、本実施形態における間接データインデックスは、同一オブジェクト内の、ある特定の他属性や、ある特定の他オブジェクトの属性を利用することにより、定義を行っている。これは、オブジェクト間での、データ部品の貼り付けや、演算によるデータ部品の共有の自動化技術に相当する。

【0213】従来の、OpenDocやOLE等の複合文章技術における発行と引用においては、ある特定のオブジェクトや属性の部分データを直接引用して、複合文章を構成する仕組みとなっている。即ち、発行元、及び引用元は、ともに固定されていたのである。

【0214】しかしながら、本実施形態は、種々のインデックス間の対応付けを、複次的に付与することのできるメタインデックスの機能を有している。このため、間接データインデックスを、さらにメタインデックスにより随時定義しておけば、メタインデックスの内容に従って、その引用先を、他のオブジェクトの定義内容に修正を加えることなく、動的かつ柔軟に変更することができる。

【0215】即ち、かかる機能を用いれば、発行や引用時における発行相手や引用相手を予め特定しなくても、後から動的に設定、変更できる仕組みが実現でき、柔軟に運用することができるという効果を奏するのである。かかる仕組みを、従来技術における発行／引用と区別するために、発信／受信と呼ぶ。

【0216】以上のように、本実施形態によれば、以下の効果が得られる。即ち、まず、データの参照時には、同一オブジェクト内の他の属性や、また、他のオブジェクトから関係を介しての、間接的なインデックスを用いることが可能となり、更に、かかるインデックスから直接データ更新を行うことも可能となった。即ち、オブジェクト間の関係もインデックス属性と同じように扱えるので、これを用いれば、他のオブジェクトの属性情報をあたかも自らの属性情報のごとくに活用が可能となる。

【0217】また、インデックスに対して複次的にインデックスを付与することができるので、抽象的でアプリケーションのビューにより近く、インスタンスの変更に影響されることのない、曖昧な、即ち柔軟なインデックスを用いることが可能となった。更に、構造化されていないデータに対しても、インデックスを付与して、データ部品としてアクセスすることが可能となった。

【0218】また、実データの変更時においても、関連オブジェクトに対し、変更情報を効率的に通知、反映し、データと関連オブジェクトの一貫性を維持することが可能となった。

【0219】さらに、新たなクラスが定義された場合で

も、新たなクラス体系において、既存のオブジェクト体系の内容を参照して仮想的なインスタンスを生成し、既存のオブジェクト情報を変換して利用することが可能となった。

【0220】また、外部アプリケーションの想定するビューが、既存のオブジェクト情報と異なっても、アプリケーションからはこれを意識せずにデータアクセスすることが可能となった。

【0221】

【発明の効果】以上説明したように、本発明によれば、以下に記載されるような効果を奏する。即ち、アプリケーションのビューにより近い、クラスレベルの抽象的かつ複次的で柔軟なインデックスの機能を提供するので、オブジェクトの変更やアプリケーションのビューの変更に影響されることなく、各アプリケーションに即して、柔軟かつ効率的に、外部からの、データおよびデータ部品へのアクセスが行えることとなる。また、これにより、オブジェクトの内容が変更された場合にも、そのメンテナンスを容易にし、外部アプリケーションから必要に応じたビューによって、効率的にアクセスすることが可能となる。

【0222】従って、図形データ、文書データ、データベースデータなど、その構造化されているデータはもとより、文章のように構造化されていないデータについても、対象データの性質を問わず、多岐にわたるアプリケーションから容易にアクセスすることが可能となる。これにより、外部からのアクセスに対するデータの独立性を高め、データ部品の共用性、再利用性を向上させることができるという効果が得られる。

【0223】尚、本発明はデータの種類自体からは透過であり、その適用対象分野は特に限定されるものではないが、とりわけ、データの変更が頻繁に行われ、かつデータ共有が強く要請される分野、即ち文書等を含むデータ編集や、設計業務などにおいて、顕著に生産性の向上をもたらすものである。

【0224】また、他の目的は、外部アプリケーションからのビューが変更、新設されて既存のオブジェクト体系との間で、不整合がある場合でも、これらの間のギャップを埋め合わせることにより、新たなアプリケーションの組み込みを容易にすることにある。

【0225】また、同一オブジェクト内の他の属性や、他のオブジェクトから関係を介しての、間接的なインデックスの機能を提供するので、関連するデータ部品の一括したアクセスやまとめた演算が行うことができる。このため、他のオブジェクトの属性情報も自らの属性情報として活用でき、データの操作時における効率の向上を図り、柔軟なデータアクセスが可能となるという効果が得られる。これにより、多岐にわたるアプリケーションからのデータのアクセスをさらに容易にし、データの共用性、再利用性を高めることが可能となる。

10

20

30

40

50

【0196】クラス間情報変換手段610は、オブジェクト管理部100で生成されるインスタンスの監視を行っており、クラス間情報変換ルールテーブルT50を参照して、ルールに該当するインスタンスが新たに生成されたときに、ルールに従って、新クラス体系における、対応するインスタンスを生成する。

【0197】即ち、既にインスタンスを持つ既存クラスから、アプリケーションのビューに対応した仮想的な新クラスを定義したときに、新たなクラスに対するインデックスが存在するかのようにより外部からアクセスできるようになる。

【0198】例えば、クラスAとクラスBの属性を再利用して新クラスCを定義し、かつ今、クラスAのインスタンスとして、a1, a2, クラスBのインスタンスとしてb1, b2が存在するとする。このときクラスcには仮想的なインスタンス

c1 = (a1, b1) から生成したもの

c2 = (a1, b2) から生成したもの

c3 = (a2, b1) から生成したもの

c4 = (a2, b2) から生成したものが存在するものとみなせる。

【0199】ここで、新しく定義した「文章管理作業」及び「全体的管理作業」クラスに対し、図30のクラス間情報変換ルールが定義されているとする。

【0200】このとき、クラス間情報変換手段610は、オブジェクト管理部100を監視して、「文章」クラスのインスタンスが生成されたとき、クラス間情報変換ルールを参照する。そして、かかるルールに従い、そのインスタンスに対する「文章管理作業」のインスタンスを新たに生成し、その属性値には、元のインスタンスのオブジェクトリファレンスを書き込む。また、「全体的管理作業」のインスタンスを生成し、これに「文書管理作業」のインスタンスを書き込む。この結果、「第1章」、「第2章」、「まとめ」、「報告書」のそれぞれに対応する文章管理作業の新インスタンスと、1つの「全体的管理作業」のインスタンスとが生成される。図31にかかる仮想インスタンスの一部を示す。

【0201】また、本実施形態においては、新しいクラスを定義した際に、その内容を一括変換する場合につき説明したが、変換は随時動的に行なわれる。例えば、新しい章＝「第4章」のインスタンスが生成されると、その進捗の情報が参照されて、新しく「第4章」に関する文章管理作業のインスタンスが動的に生成される。

【0202】また、前述のメタインデックス管理手段440を用いれば、例えば新クラス管理手段620で、新しいクラスを定義した際の、新しい属性に対応する情報を既存オブジェクトから抽出する場合に、既存オブジェクトの属性に適切なものがあればこれを検索して、クラス定義の間接データインデックス属性定義として組込むことができる。

【0203】次に、外部アプリケーションから、本実施形態に係る情報管理装置の有するデータに、柔軟にアクセスする場合の例として、プラグイン部700の処理内容を以下に説明する。

【0204】プラグイン部700は、外部アプリケーションが想定するビュー、即ち、クラス名や属性名等の用語や、クラス体系、データ構造等が、既存のオブジェクトの用語やクラス体系、属性構造等と本来整合性がない場合においても、メタインデックスや、クラス間情報管理部600と連動して、かかる整合性を維持して、外部アプリケーションが、既存のデータに仮想的にアクセスすることのできる機能を提供するものである。

【0205】まず、用語の違いがある場合には、メタインデックステーブルT40に、用語の違いの関連づけ、即ち、アプリケーション側の用語と、オブジェクト側の用語の対応を登録する。かかるメタインデックス定義により、外部アプリケーションは、既存のデータの用語との違いを意識することなく、データアクセスをすることができる。

【0206】また、クラス体系やデータ構造（オブジェクト側という属性構造）に違いがある場合には、新クラス管理手段620に、新クラス体系を定義し、また、クラス間情報変換ルールテーブルT50に、変換ルールを定義する。かかる定義により、クラス間情報変換手段610が、新しいクラスと既存のクラスの属性間のデータの整合性を、逐次維持するので、外部アプリケーションは、既存のデータベースの内部構造を知らなくても、容易に既存データへのアクセスを行うことができる。

【0207】またこのとき、上書き手段430により、書き込みもできるので、単なる間接参照による擬似データ構造への更新をするのではなく、実際にデータベースの実データへの書き込みができる。

【0208】最後に、オブジェクトの内容を表示する処理について、以下に説明する。

【0209】入出力部800中の表示手段820の管理するレイアウト定義テーブルT140には、オブジェクトの内容が、具体的な値のかわりにインデックスにより記述されている。ここで、表示を行う際には、かかるインデックスの値を、メタインデックス管理手段440に渡すことにより、対応する属性値を取得し、インデックス表示にかかる属性値で置き換えて最終的な表示内容の合成を行う。

【0210】レイアウト定義は、例えばHTML (Hyper Text Markup Language) 言語で行ない、この中でインデックスを利用した文字列により、オブジェクトの情報を組込んだフォーマットで定義される。

【0211】オブジェクト情報の表示は、レイアウト定義中のインデックス文字列を、入出力手段810の関数getValue (インデックス文字列)

を利用することにより、実際の値へと展開する。ここ

【0180】まず、データアクセス部300は、属性定義情報を参照し、「第1章」の属性＝”要約”のアクセス関数の情報を得る(S500)。

【0181】次に、得られたgetValueがオブジェクトに対する読み込み型の関数であることから、属性＝”要約”の属性情報の中から読み込み型のアクセス情報を得る(S510)。

【0182】{doc, read, youyaku, section1.doc}そこでdccに対する読み込みに関するアクセス関数を、図25に示すアクセス関数テーブルを参照して、結果として関数ポインタ\*youyakuを得、以下の関数を合成する(S520)。

【0183】functionCall (\*youyaku, section1.doc)

データアクセス部300は、かかる関数処理を、データベースに対して実行することにより、データsection1.doc内部にアクセスして、section1.docの内容のうちから、要約のみを、データ部品として得る(S530)。

【0184】この場合、結果として“はじめに”をfunctionCallの関数値として得るので、この値をgetValueの値として、入出力部800を通じてユーザーに対して表示する(S540)。

【0185】第2に、データの書き込みの場合について説明する。

【0186】データに対する書き込みの場合には、属性定義情報のうち、書き込みに関する属性定義が参照されて実行される。例えば、入出力部800から以下の書き込み要求が入力されると、

putValue(「第2章」. 内容, “特許明細書における項目の構成と...”)

データ書き込みの場合に説明したのと同様のフローに従い(S500～S530)、以下の関数、

functionCall (\*PutNaiyo, section2.doc, “特許特許明細書における項目の構成と...” ) が合成、実行されて、属性値記録テーブルT20、及びデータベースにおけるsection2.docの内容が更新されることとなる。

【0187】次に、実データが変更された場合の、変更管理部500による、関連オブジェクトの自動更新の処理の内容につき説明する。変更管理部500は、あるオブジェクトの属性値に、書き込みによる変更が発生した場合に、かかる変更を関連する別の属性に反映させる機能を提供する。尚、ここでは実データへの変更として説明を行うが、属性値定義テーブルT20上の属性値データに対する変更も、実データへの変更と同期をとって行われるため、実際の変更対象にはかかる実データも含まれることとなる。図27は、かかる変更管理部500の処理の流れを示すフローチャートである。

【0188】ここでは、「第1章」の行数が、10から20に増えた場合につき、図27に基づき、具体的に説明する。まず、変更検知手段510(pub\_no1)

は、変更検知テーブルT110に登録されたオブジェクトの内容の変更を監視しており(S400)、かかるデータ変更が生じた場合(S410)、これを検知し、変更された内容(20)を、変更通知手段520(med\_no1)に送信する(S420)。

【0189】変更通知手段520は、変更反映のために必要となる他の参照データとして、変更通知元モジュール名から変更通知テーブルT120を参照して他の参照データとして、第2章(pub\_no2)、まとめ(pub\_mat)の行数を得て(S430、S440)、これらを計算式に従い和計算して新しい値70を得る(S450)。この得られた値を、変更反映手段530(sub\_no1)に通知すると(S460)、変更反映手段530は、変更反映テーブルT130を参照して、反映対象である「報告書」の、属性＝”総行数”の属性値を更新する(S470)。

【0190】変更管理部500では、オブジェクト管理部100におけるインスタンス生成と連動して、関連するモジュールの生成と、オブジェクトへの組み込みが実行される。

【0191】尚、変更管理部500を用いることにより、複数のオブジェクトが同一の属性値を参照している場合には、変更検知手段510に相当するモジュールは一つのまま、その通信先を複数、変更検知テーブルT110に登録するだけで、データ変更が生じた場合、同時に複数の該当するオブジェクトに反映することが可能となり、効率の良い変更管理が実現されることとなる。

【0192】尚、変更管理部500のモジュールの生成、及び組み込みは、インスタンス生成時に行われ、後の変更の検知、通知及び反映は、データの変更が生じた場合に随時行われる。

【0193】次に、新しいクラスが定義された場合における、クラス間情報管理部600の処理の内容につき、以下に説明する。まず、新しいクラスが必要となった場合には、新クラス管理手段620において、新しいクラス情報を追加定義、登録する。この場合に、オブジェクト管理部100に既に存在するクラス情報を利用して、クラス定義を記述することができる。図29に、本実施形態における新クラス定義の一例を示す。

【0194】本実施形態に係る情報管理装置を、既に運用している途中において、既存のクラス定義の体系と異なる、新たな新たなクラス定義の体系を用いる、新たなアプリケーションのビューにより、情報管理装置の有するデータにアクセスするためには、以下の手順に従う。

【0195】即ち、まず、入出力部800等のインターフェースを用いて、新クラス定義手段620に新しいビューに対応するクラス群の定義を行う。更に、クラス間情報変換手段610の管理するクラス間情報変換ルールテーブルT50に対して、新クラス体系と既存のクラス体系との関連づけを記述したルールを登録する。

同時に行うことで、他からのアクセスに対してのデータの一貫性が維持されることとなる。

【0165】また、インデクス管理手段410は、属性値型が間接データインデクスであった場合には、上書き手段430に制御を渡し、上書き手段430は、属性定義テーブルT10上の間接データインデクスの計算式から、上書き可能と判断された場合は（S217）、さらに計算式を用いて書き込む値の計算を行い（S218）、一方、上書き不可である場合には、エラー値を返して終了する（S220）。書き込む値を戻されたインデクス管理手段410は、格納場所記述から、データインデクスの場合において説明したのと同様の属性値の書き込みを行う（S219）。

【0166】インデクス管理手段410は、また、属性値型が、データインデクスでも間接データインデクスでもない場合には、上書き可能か否かの判断を、属性値記録テーブルT20の該当エントリーの上書き可能か否かを示すフラグを参照することにより行い（S215）、上書き可能であれば、格納場所記述から、上記と同様の属性値の書き込みを行い（S219）、上書き不可である場合には、エラー値を返して終了する（S220）。

【0167】以下に、具体的な値を用いて上述の処理の流れを説明する。いま、

putValue（「第1章」．コメント，“この章には大幅修正が必要”）が入出力部800から与えられた場合、メタインデクス管理手段440は、“「第1章」．コメント”に対応するメタインデクスが存在しないので（S200）、“「第1章」．コメント”を、一次レベルのインデクスとして、値のアクセスを行う。インデクス管理手段410は、オブジェクト管理部100を介して、属性＝”コメント”を記録するオブジェクト＝「第1章」を得（S211）、また、格納場所管理テーブルT30を参照することで、格納場所記述＝”第1章#コメント”を得る（S212）。また、さらに属性定義テーブルT10を参照することにより、“第1章．コメント”の属性型が単値で、属性値型がSTRINGであることを得る（S214）。

【0168】この場合、属性値型が間接データインデクスでも、データインデクスでもなく、上書き可能であれば、属性値＝“この章には大幅修正が必要”の書き込みを、属性値記録テーブルT20およびデータアクセス部300を介してデータベース上の実データに対して行う（S219）。図24に、書き込み後の属性値記録テーブルT20の内容の一部を示す。

【0169】第二に、オブジェクトの属性値の読み込みを要求した場合につき、図22および図23を示して、書き込み処理と相違する部分のみ説明する。

【0170】入出力部800から以下の読み込み関数が呼び出されると、

getValue（インデクス記述）

書き込みの場合と同様に、メタインデクス管理手段440に制御が渡り、図22に示すメタインデクスの検索処理が行われ、実際にアクセスする一次レベルのインデクスを得る（S300）。

【0171】次に、得られたインデクスから、インデクス管理手段410を用いて、インデクスに対応する値を得るのである（S310）。

【0172】図23は、かかるインデクス管理手段410における、属性値取得のさらに詳細を示すフローチャートである。

【0173】インデクス管理手段410に制御が渡ると、書き込みの場合と同様に、検索されたオブジェクト名、属性名から、格納場所記述を取得し（S311、S312）、属性値型の判断を行う（S314）。この場合、属性値型が間接データインデクスであった場合には、更に、間接データインデクス管理手段420が、属性定義テーブルT10の該当する計算式を参照して、属性値の計算を行い、値を返す（S316、S317）。

【0174】一方、属性値型が、間接データインデクス以外の、データインデクスまたはその他であった場合には、格納場所記述を用いて、属性値の読み込みを行うのである。かかる読み込みは属性値記録テーブルT20とデータベース上の実データとの同期が保持されているならば、いずれから読み取りを行っても任意であるが、属性値記録テーブルT20からの読み込みを行った方が、パフォーマンス上の観点からは望ましい。

【0175】今、以下の関数が具体的な値とともに与えられたとすると、

getValue（「第1章」．コメント）

上述の処理の流れ（S300～S317）に従って、値＝“この章には大幅修正が必要”を得ることとなる。

【0176】尚、メタインデクスを用いた検索を行なうに当たっては、インデクス間の整合性を計算する機能が必要であるが、これには、例えばパターンマッチング等の技術を用いることができる。

【0177】ここで、属性値型が、データインデクスであった場合のアクセスの詳細をさらに、説明する。

【0178】第1に、データの読み込みの場合について説明する。いま、入出力部800を通じ、以下の関数がメタインデクス管理手段440に対して入力されたとする。

【0179】getValue（「第1章」．要約）

ここで、“「第1章」．要約”は、メタインデクステーブルT40に登録されていないため、与えられたインデクス記述をインデクス管理手段410に渡して、属性値の参照が行われるが、“第1章”．要約に対応する属性の属性値型がデータインデクスなので、データアクセス部300に制御が渡って、以下のアクセス手順に従い、アクセスが実行される。図32に、かかるアクセスの処理手順を示す。



データインデクス以外の、通常の属性値のアクセスと同じ手順により値を得る方法も可能である。

【0147】次に、間接データインデクスにおけるアクセス権が、書き込み可能であった場合の、データの書き込みの手順について、以下に説明する。

【0148】間接データインデクス定義では、参照先のインデクスのアクセス権と演算の性質から、定義された間接データインデクスのアクセス権を計算する。

【0149】そして、間接データインデクスによるアクセスでも、アクセス権が書き込み可能な場合には、これをおこなうことができる。かかる上書き操作は、上書き手段430が、まず属性が書き込み可能か否かを、属性定義の計算式の逆関数があるか否かにより判断を行い、逆関数があれば書き込み可能なものとして書き込むべき値を計算し、かかる値を、データアクセス部300が、図16のステップS110により得られた格納場所指定に対して、実際のデータベースに書き込みを行う。また、後述する変更管理部500の中に、上書き用のモジュールを追加することによっても、上記上書き処理は可能となる。

【0150】次に、間接データインデクスの示す対象のオブジェクトが複数の場合、即ち、張り込み先の属性型がテーブルである場合のデータアクセスの例を、以下に説明する。

【0151】今、「報告書作成管理」は「管理作業」クラスのインスタンスで、属性＝”進捗”は、図17に示すように定義されている。

【0152】ここで、「報告書作成管理」、関連文章の値はリストであり、それぞれのリスト要素を行として、このリスト要素の属性＝”名前”と、属性＝”行数”の属性値が列に張られている。

【0153】この時、属性値記録テーブルT20には、図18に示す内容が記録されている。

【0154】また「報告書作成管理」オブジェクトには、テーブルのkey属性を基にインデクスと格納場所との関連付けが、図19に示すように格納場所管理テーブルT30に記録されている。

【0155】この場合においても、具体的な間接データインデクスによるアクセスの実現方法は、アクセス時に関連する情報をたどって値を得る方法と、変更管理部500によるデータ変更時の変更管理による方法がある。

【0156】次に、ユーザーが、入出力部800を用いて、オブジェクトの属性値、または実データにアクセスする場合の処理内容につき、以下に説明する。

【0157】ここで、入出力部800におけるオブジェクトへのアクセス関数としては、  
getValue (インデクス記述)  
putValue (インデクス記述、値)  
の2種類がある。

【0158】図20乃至図23に、入出力部800から

オブジェクトの属性値にアクセスした場合の処理のフローチャートを示す。尚、図20乃至図23に示すかかるアクセスの処理は、入出力部800からの要求に限られず、外部アプリケーションからプラグイン部700を経由した場合でも、同様の処理が行われる。

【0159】第1に、オブジェクトの属性値の書き込みを要求した場合につき、図20および図21を示して説明する。

【0160】入出力部800から以下の書き込み関数が呼び出されると、

putValue (インデクス記述、値)

まず、メタインデクス管理手段440に制御が渡り、図20に示す処理が行われる。メタインデクス管理手段440は、最初に、与えられたインデクス記述が、メタインデクスであるか否かの判断を行う。即ち、メタインデクステーブルT40を参照して、入力されたインデクス記述に対応するテーブルエントリがあるか否かを検索し、あった場合には、実際にアクセスする一次レベルのインデクスを得る(S200)。この際、与えられたメタインデクスが、メタインデクスに対してさらに付与されたものであった場合には、再帰的にテーブルエントリの検索を行うことで一次レベルのインデクスを得ることとなる。次に、得られたインデクスから、インデクス管理手段410を用いて、インデクスに対応する値を得るのである(S210)。

【0161】図21は、かかるインデクス管理手段410における、属性値取得のさらに詳細を示すフローチャートである。

【0162】インデクス管理手段410に制御が渡ると、まず、与えられたインデクスから、オブジェクト管理部100が保持する、属性定義テーブルT10及び必要に応じて関係定義テーブルT80を参照して、関係を手繰ることにより、インデクスに示される属性を直接記録するオブジェクト名および属性名を取得する(S211)。さらに、得られたオブジェクト名及び属性名から、格納場所管理テーブルT30を参照して、該当属性の格納場所記述を取得する(S212)。

【0163】更に、インデクス管理手段410は、属性値型を判断して、それぞれに対応する処理を行う。即ち、オブジェクト管理部100が保持する属性定義テーブルT10を参照して、与えられたインデクスが、データインデクスか、間接データインデクスか、それ以外なのか、の属性値型を得る(S214)。

【0164】ここで、属性値型がデータインデクスである場合には、さらに上書き可能な属性であれば(S215)、格納場所記述からデータアクセス部300を介して、属性値の書き込みを行い(S219)、一方、上書き不可である場合には、エラー値を返して終了する(S216)。尚、かかる属性値の書き込みは、データベース上の実データ及び属性値記録テーブルT20に対して

含むメタインデクステーブルT 4 0の一部を示す。

【0 1 2 9】メタインデクス管理手段4 4 0は、与えられたメタインデクス記述に対し、メタインデクステーブルT 4 0から該当する一次レベルのインデクスを検索する。今、メタインデクス”文章かつ結論”を与えたとすると、以下の関数処理により、searchIndex (“文章かつ結論”) 値として“まとめ、要約”が得られる。

【0 1 3 0】また、メタインデクスの定義は柔軟性に富んでおり、定義の仕方によっては、複数のインデクスを集めて、目的に合うように合成する等も可能である。

【0 1 3 1】例えば、“進捗に関する行数/進捗に関する予定行数”というメタインデクス記述に対して該当する一次レベルのインデクスを検索すると、進捗に関連する行数には、第1章、行数 進捗に関連する予定行数には、第1章、作成、目標行数を得るので、これらを組み合わせて計算を行ない、以下の結果を得ることができる。

【0 1 3 2】searchIndex (“進捗に関する行数/進捗に関する予定行数”) = 第1章、行数/第1章作成、目標行数

次に、初期登録処理の後に、間接データインデクスの追加を行った場合の処理の一例につき、以下に説明する。

【0 1 3 3】今、オブジェクト=「報告書」に対して、新たな属性=“責任者”を付与した場合、オブジェクト=「報告書」に対応する属性定義テーブルT 1 0の内容が、図1 4 (a) に示すエントリーが追加されて書き変わると同時に、既に生成されているオブジェクトの属性構造も変更される。

【0 1 3 4】ここで、「報告書」における属性=“責任者”の属性値型は、間接データインデクスで、{作成者}は、他の属性を示している。

【0 1 3 5】また、オブジェクト=「報告書」に、同様に間接データインデクスを属性値に持つ、属性=“総行数”を新たに追加するとすると、属性定義テーブルT 1 0の内容が、図1 4 (b) に示すエントリーが追加されて書き変わる。

【0 1 3 6】ここで、「報告書」と子文章の関係にあるのは複数の「文章」のオブジェクトであり、“子文章、行数”はリストとなる。SUMはリストに対する総和の演算子である。これは属性=“子文章”で参照される全てのオブジェクトに対して、属性=“行数”を参照しこれらの総和を計算して、属性=“総行数”の値とすることを示している。

【0 1 3 7】次に、以下に、本実施形態を用いた、データまたはデータ部品へのアクセスについて、その処理内容を説明する。

【0 1 3 8】まず、間接データインデクスを用いた場合のデータアクセスについて説明する。

【0 1 3 9】図1 6に間接データインデクスを用いたア

クセスのフローを示す。尚、以下の処理中において、途中で複数のオブジェクトが得られる場合は格納場所をリストとして保持し、複数のオブジェクトの対応する場合が複数回発生する場合には入れ子構造を持つリストとして保持する。

【0 1 4 0】第1の例として、「報告書」における属性=“責任者”の例で間接データインデクスのアクセスを説明する。上述した間接データインデクスの追加において、図1 4 (a) に示したように、属性=“責任者”の属性値型は、間接データインデクスであり、定義内で同オブジェクトにおける、別の属性=“作成者”を参照している。間接データインデクス管理手段4 2 0は、間接データインデクス定義=“責任者は{作成者}です”から、参照すべきインデクスとして、“報告書、作成者”を得る(S 1 0 0)。

【0 1 4 1】次に、インデクス管理手段4 1 0を参照し、“報告書、作成者”は「D氏」であることを得る(S 1 2 0)。尚、この際の処理をさらに具体的に詳述すると、まず、インデクス記述を左から順にスキャンし、最終的にデータを保存する属性を特定する(S 1 1 1)。次に、格納場所管理テーブルT 3 0を参照して、対応する格納場所指定を得(S 1 1 2)、かかる格納場所指定にアクセスして対応する属性値を得るのである。

【0 1 4 2】最後に、属性値にD氏を埋め込み、結果として“責任者はD氏です”を得る(S 1 3 0)。

【0 1 4 3】次に、第2の例として、オブジェクト=「報告書」に対する間接データインデクスの属性値を持つ属性=“総行数”を参照する。

【0 1 4 4】+総行数、単値、IdIndex, “SUM ( {子文章、行数} ) ”

この場合は、「報告書」と子文章の関係にあるのは、複数の「文章」のオブジェクトであり、従って、子文章、行数はリストとなる。SUMはリストに対する総和の演算子である。つまり、上記の計算式は、属性=“子文章”で参照される全てのオブジェクトに対して、属性=“行数”を参照し、これらの総和を計算して属性=“総行数”の値とすることを示している。

【0 1 4 5】ここで、報告、総行数を参照すると、全ての子文章、行数が参照されて(S 1 2 0)、これらの和が計算されたものが値として返される(S 1 3 0)。

今、

第1章、行数=1 0、

第2章、行数=3 0、

まとめ、行数=2 0、

であるとして、“報告書、総行数”=6 0になる。

【0 1 4 6】間接データインデクスに関するアクセス方法は、上述した手順によるものに限定されるものではない。他の方法としては例えば、変更通知手段5 2 0により、同インデクスで示される属性に対し、変更値をデータ変更の度毎に反映させておいて、これを用いて、間接

的には、データ部品のアクセスを行うための初期登録内容、間接データインデクスを用いたデータアクセス、メタインデクスを用いたデータアクセス、実データが変更された場合の関連オブジェクトへの反映、外部アプリケーションが追加・変更された場合の新しいアプリケーションからのデータアクセス、および管理されているオブジェクトの内容表示等につき、順に以下に説明する。

【0111】まず、初期登録としては、データ部品にアクセスするためのオブジェクト環境を設定する。即ち、まずオブジェクト管理部100において、オブジェクトの初期化を行なう。

【0112】オブジェクト管理部100は、初期化用の関数列を記録した、オブジェクト初期化ファイルを参照してオブジェクトを生成する。オブジェクト管理部100には、予め初期化用の関数が用意される。図8に、オブジェクト初期化ファイルの一例を示す。本実施形態においては、

オブジェクト作成関数：objectCreate（クラス名，インスタンス名）

オブジェクト関係挿入関数：relationAssert（オブジェクト名1，オブジェクト名2，関係ロール名）

という2種類の関数を用いる。これらの関数を初期化ファイルの上から下に読み込んで、オブジェクトの作成とオブジェクト間の関係付けとを順次実行する。

【0113】このとき作成されたインスタンスには、属性＝”名前”としてインスタンス名が記録される。

【0114】また、この際に生成されたオブジェクトには、前述したように、必要なデータインデクス、及び間接データインデクスが、すでにそれぞれの属性値型により、定義がなされている。

【0115】インスタンス生成時には、更に、属性定義テーブルT10が生成され、クラス定義から属性情報がコピーされることにより、個々のインスタンスの属性定義が記録保存される。

【0116】次に、生成されたオブジェクトに対して、データ登録部200により、データを登録する。本実施形態では、以下の登録用の関数を用いて、文章ファイルを登録する。

【0117】dataRegistor（インデクス，データオブジェクト）

かかる登録処理は、初期化ファイルに関数を記述することによりバッチ的に行ってもよいし、ユーザーが、入出力手段810を通じて、任意のタイミングで指示してもよい。図9に、データ登録内容の一例を示す。

【0118】かかるデータ登録により、属性値記録テーブルT20にオブジェクトの属性値が記録される。図13に、データ登録後の、属性値記録テーブルT20の内容の一例を示す。属性値記録テーブルT20内では、承継された属性も含めて記録されており、また、間接データインデクスの属性値には、参照値を埋め込んだものを

記録している。かかる間接データインデクスの属性値を参照するためには、インデクスや属性に関する定義情報に基づいて、データアクセス部300を用いて、実データにアクセスしてもよい。また、保有する元のデータの変更時に、変更管理部500が、属性値記録テーブルT20の内容を同期的に更新する構成とすれば、ユーザーは属性値記録テーブルT20の内容のみを参照することで、高速に属性値にアクセスすることが可能となる。

【0119】また、このデータ登録時に、格納場所管理テーブルT30に、定義したオブジェクトの属性毎に、対応する格納場所の登録を行う。

【0120】さらに、データアクセスに必要な、データアクセス手段をデータアクセス手順格納テーブルT90に、アクセス関数をアクセス関数テーブルT100に、それぞれ登録を行う。

【0121】次に、他のオブジェクトから間接的に属性を参照するために必要となる、関係の挿入を行う。図10に、本実施形態におけるオブジェクト間の関係を挿入する処理手順を示す。以下このアルゴリズムに従って以下の処理の動作を説明する。下記に示すのが、関係を挿入するために用いられる関数である。

【0122】relationAssert（「報告書」，「第1章」，親文書）

まず関係定義テーブルT80に、図11（a）に示す関係を登録する（S10）。次に、元オブジェクトにおける属性＝後ロール名に、図11（b）に示す先オブジェクトのリファレンスを追加する（S20）。

【0123】次に、先オブジェクトにおける属性＝前ロール名に、図11（c）に示す元オブジェクトのリファレンスを追加する（S30）。

【0124】続けて、属性間制約＝“{章目}．報告書名<＝{報告}．名前”を挿入する。

【0125】この場合の制約は、章目の属性＝“報告書名”は報告の属性＝“名前”を参照していることを示している（S40）。

【0126】最後に、関係付けに関連するオブジェクトが取得できるように、格納場所情報を更新する（S50）。

【0127】図12に、更新された「報告書」の、格納場所管理テーブルT30の一部を示す。尚、同様に、関係先の「第1章」についての格納場所管理テーブルT30のエントリーも更新される。

【0128】次に、アプリケーションのビューの新規追加や、変更に対応して任意に設けることのできるメタインデクスの登録を行う。かかる登録は、メタインデクステーブルT40に対して、メタインデクスと、その基礎となるインデクスの対応関係のエントリーを追加することにより行う。例えば、“まとめ．要約”という属性に対して“重要，要約，結論，文書”というメタインデクスをつけることができる。図15に、かかる登録内容を

更が生じた場合に、対応するオブジェクトの属性値を更新すると同時に、更に、間接データインデクスにより間接的に参照している他の属性も併せて更新し、変更に関する管理を行う。変更管理部 5 0 0 は、下位構成として、更に、データの変更の監視を行う変更検知手段 5 1 0、データ変更が生じた場合に関連する属性に対する変更値を計算し、これを通知する変更通知手段 5 2 0、関連するオブジェクトの属性値の更新を行う変更反映手段 5 3 0 とを有している。

【0 1 0 0】変更検知テーブル T 1 1 0 は、変更検知手段 5 1 0 が管理を行い、監視対象オブジェクトおよびインデクス、変更通知手段モジュール名が記録されている（図 2 8（a））。

【0 1 0 1】変更通知テーブル T 1 2 0 は、変更通知手段 5 2 0 が管理を行い、変更通知元、変更反映モジュール名、データ変更の反映に必要な計算用参照データおよび計算式が記録されている（図 2 8（b））。

【0 1 0 2】変更反映テーブル T 1 3 0 は、変更反映手段 5 3 0 が管理を行い、反映対象オブジェクト及びインデクス、変更通知手段モジュール名が記録されている（図 2 8（c））。

【0 1 0 3】次に、クラス間情報管理部 6 0 0 は、外部アプリケーションの用いる新しいビューに対応する新しいクラスの定義情報、及びこれと既存のクラス定義との間の変換ルールの管理を行う。クラス間情報管理部 6 0 0 は、下位構成として、更に、新しいクラスの定義及び定義情報の管理を行う新クラス管理手段 6 2 0 と、新しいクラス定義と、既存のクラス定義との間の変換ルールを管理し、かかるルールに基づいて、オブジェクト管理部 1 0 0 を監視して、既存クラスのインスタンスに対応する新クラスの仮想インスタンスを漸次生成するクラス間情報変換手段 6 1 0 とを有している。

【0 1 0 4】クラス間情報変換ルールテーブル T 5 0 は、クラス間情報変換手段 6 1 0 が管理を行い、かかるクラス間における変換ルールの定義情報が記録されている。

【0 1 0 5】次に、プラグイン部 7 0 0 は、すべての外部アプリケーションとのインターフェース部分である。そして、本実施形態に係る情報管理装置の保有する実データを利用する外部アプリケーションに対して、そのアプリケーションが新たなビューを用いてアクセスする場合、即ち、そのアプリケーションが想定するデータの用語等が、オブジェクト管理部 1 0 0 の管理する属性等の用語と整合しない場合や、外部アプリケーションが想定するクラス体系やデータ構造が、オブジェクト管理部 1 0 0 の管理するクラス体系や属性構造と異なる場合にも、実データの利用を可能とする。即ち、用語が相違する場合には、メタインデクス管理手段 4 4 0 に対してかかる用語の相違をメタインデクスとして登録を行い、また、クラス体系や属性構造が相違する場合には、アプリ

ケーションが想定するクラス体系を、新クラス管理手段 6 2 0 に定義し、さらに新クラスと既存のクラスの間の変換ルールを、クラス間情報変換手段 6 1 0 に登録することで、アプリケーション組み込みにおける整合性を維持するのである。

【0 1 0 6】入出力部 8 0 0 は、本実施形態に係る情報管理装置の有するオブジェクトの各種情報を、要求に応じて表示する。

【0 1 0 7】入出力部 8 0 0 は、更に、下位構成として、オブジェクトの内容の参照、書き込みなどのアクセスを行う入出力手段 8 1 0 と、実際のオブジェクトの内容の表示を、定義したレイアウトに従い表示する表示手段 8 2 0 とを有している。レイアウト定義テーブル T 1 4 0 は、表示手段 8 2 0 が管理を行い、表示用レイアウトの定義情報が記録されている。

【0 1 0 8】尚、ここで、オブジェクトの管理は、オブジェクトデータベースに代表されるオブジェクトを取り扱えるデータベースでも実現可能であり、従来の関係データベースに必要な機能を付与することによりを実現することもできる。また、本実施形態における実データとは、データベースに格納されているファイルや表データに代表されるコンピュータ上の資源としての実体を指し、何らかの指示情報（固有の名前等）により、ソフトウェアから容易に参照できるものとする。即ち、その編成は任意であり、構造化されたデータでも、構造化されないフラットなデータの状態でもよい。また、実データが保存されるデータベースは、関係データベースでもよいし、ファイル保存でもよい。さらに、オブジェクトやデータを保存するデータベースの所在は、集中的に、単一のコンピュータで管理されていてもよいし、ネットワークによって互いに接続する複数のコンピュータ上に分散して管理されていてもよい。すなわち、例えばオフィス環境にて個人個人の専用コンピュータ上の特定のデータに対してオブジェクトを生成し、これをネットワークを通じて他の利用者と交換するといった運用をすることもできる。この場合、オブジェクトの分散管理に関しては、既存技術である分散オブジェクト技術を利用する。

【0 1 0 9】また、ここで、本実施形態が具備するハードウェア構成は、各種処理を行うための CPU と、キーボード、マウス、各種リーダー等の入力装置と、メモリ装置やディスク装置等の外部記憶装置と、ディスプレイ装置、プリンタ装置等の出力装置とを備えた通常のコンピュータシステムの他、オブジェクト処理に適した専用ハードウェアを用いても実現することができる。尚、前記 CPU は、各種処理を行う演算部と、前記処理の命令及びデータを記憶する主記憶部とを具備する。

【0 1 1 0】本実施形態は、上記のように構成されており、以下、本実施形態に係る情報検索方法を用いて、保有するデータのデータ部品毎のアクセス、再利用を行う場合の処理内容について、図面に基づき説明する。具体

どり着くことを示している。例えば、  
 “「報告書」. 子文章. 作成者. 年齢”  
 は

“オブジェクト「報告書」の子文章の作成者の年齢”  
 を指し示している。

【0076】尚、間接データインデックスの場合のインデックス記述においては、自分自身をデフォルトのオブジェクトとするため、その記述中には、オブジェクトを含まない、相対的なインデックス記述となる。

【0077】ここで、インデックス記述に関係ロール名を用いた場合には、一般的に1つの関係ロールには複数のオブジェクトが関連するため、1つのインデックス記述の指示する属性は複数になる。

【0078】例えば、「報告書」についての子文章が複数ある場合には、

属性= “「報告書」. 子文章. 作成者. 年齢”  
 は複数の作成者の年齢の集合を示すことになる。

【0079】本実施形態においては、このように、1のインデックス記述に複数の値が対応する場合に、その中の特定の属性を指示する記述方法を有している。具体的には、以下の3つの記述方法を用いる。

【0080】第1に、番号により指定する記述方法である。

【0081】“「報告書」. 子文章# 2. 作成者. 年齢”

は“「報告書」. 子文章”の2番目の「文章」のオブジェクトを特定指示している。

【0082】第2に、属性の値により指定する記述方法である。

【0083】“「報告書」. 子文章. {名前=第3章}. 作成者. 年齢”

は名前が第3章である関連文書を特定している。

【0084】第3に、キー属性により指定する記述方法である。

【0085】属性定義において、キー属性が指定されている場合（この場合は名前）にはキー属性値を用いて以下のような指定ができる。

【0086】“「報告書」. 子文章. 第3章. 作成者. 年齢”

これらのインデックス情報の関連付けは、格納場所管理テーブルT30に記録され、インデックスを解釈する際に活用される。

【0087】尚、クラス定義におけるアクセス権とは、管理上のアクセスの許可に関する情報や実際の書き込み、読み込み可能性に関する情報などを示し、属性毎に所持者が異なってもよい。また、さらに、その属性値の所持者に関する情報等を含んで、アクセス情報として定義してもよい。

【0088】次に、図6に、本実施形態における、クラス管理テーブルT60の内容を示す。

【0089】「万物」は全てのオブジェクトの親クラスであり、属性として“名前”と、管理するテーブルのリファレンスである“データ”をもつ。全てのクラスは「万物」のサブクラスであり、「万物」の属性である“名前”と“データ”とを継承する。図6では、便宜のため継承される属性定義については省略している。

【0090】「文章」クラスは「万物」クラスのサブクラスであり、属性=“タイトル”は、間接データインデックス属性定義がされており。その値は属性=“名前”を参照することを示している。属性=“作成者”は、関係ロール名であり、その値は関係のある他のオブジェクトリファレンスとなっている。

【0091】「章目」クラスは「文章」クラスのサブクラスであり、属性=“内容”はデータインデックス属性定義で、属性値として、

“{doc, read, GetNaiyo, null}, {doc, write, PutNaiyo, value}”

を持つ。これは、読み込み用には、

“タイプ=doc, アクセス関数=GetNaiyo, 引数=なし”

の関数が指定されており、書き込み用には、“タイプ=doc, アクセス関数=PutNaiyo, 引数=あり”が指定されていることを示している。

【0092】属性=“行数”、“要約”も、同様にデータに対するインデックスとして定義されている。

【0093】属性=“作成者名”は間接データインデックス定義がなされており、属性=“作成者”から参照されるオブジェクトの属性=“名前”を参照している。属性=“親文章”は、関係ロール名である。

【0094】次に、「管理作業」クラスでは、テーブルの属性型が使用されている。

【0095】すなわち、属性=進捗は、属性値型が間接データインデックスで、その属性型はテーブルである。

【0096】+進捗テーブル、IdIndex, “関連文章.< {名前=key, 行数} >”

この間接データインデックス値の意味は、属性=“関連文章”が示す複数のオブジェクトに対してテーブル行を展開し、さらにそれぞれの行に対して、<名前, 行数>という属性の順番で列展開を行い、テーブルを構成することを示している。

【0097】本実施形態では、便宜的に関連文章リスト中の番号を利用してインデックスを記述することとすると、属性=“進捗”の値としては、関連文章が3つあるとすれば、図7に示すテーブルに展開されることとなる。尚、属性値にあるkeyとは、“名前”が関連文章を区別するためのキー属性であることを示している。

【0098】尚、図4に、図6のクラス定義に対応して、関係管理手段130によって定義された関係定義テーブルT80の内容を示す。

【0099】次に、変更管理部500は、実データに変

0は、オブジェクトにおける、間接データインデクスの定義、およびかかる間接データインデクスが示すデータへの参照、書き込み等のアクセスを行う。尚、間接データインデクスの定義については、後述する。

【0060】上書き手段430は、オブジェクトの属性が書き込み可能かどうかについての情報を記録し、該当属性が書き込み可能な場合には、書き込むべき属性値の計算を行う。

【0061】メタインデクス管理手段440は、オブジェクトに対する一次レベルのインデクスに対して、更に複次的に付与したインデクス情報であるメタインデクスの管理を行い、与えられたメタインデクスを用いた記述に対し、オブジェクト管理部100に管理されているオブジェクトから、該当するインデクスを持つものの検索を行う。尚、ここでメタインデクスとは、いわば、外部アプリケーションのビューに対応した、抽象レベルのインデクスであり、その基礎とするインデクスは、一次レベルのインデクスの他、メタインデクスに対して更に複次的にメタインデクスを付与することも可能である。また、ここで、一次レベルのインデクスとは、データ部品を直接又は間接に指し示す、データインデクスまたは間接データインデクス自体とは異なり、かかるインデクスが付されて格納されたオブジェクトに対して、アクセスを行うためのインデクス記述をいう。

【0062】メタインデクステーブルT40は、メタインデクス管理手段440が管理を行い、かかるメタインデクスと、一次レベルのインデクス記述との対応関係が記録されている。

【0063】ここで、オブジェクトのクラス定義における属性情報の詳細について、図5を用いて以下に説明する。図5に示すように、クラス定義における属性情報には、具体的には、属性名と属性型と属性値型がある。

【0064】属性型は属性の値の入れ物としてのデータ構造を示し、例えば図5に示すように単値、リスト、テーブルといった型がある。ここで単値 (Singlevalue) は属性値が1つのみ格納される型、リスト (List) とは複数の属性値が記録される属性型で一つ自然数により唯一の属性値が参照できる型、テーブル (Table) とは複数の属性値が記録される属性値で二つの自然数の組み合わせに対して唯一の属性値が参照できる型である。尚、クラス定義においては、更に、当該属性の属性値に対して許される読み込み、書き込み等のアクセスの種別がアクセス権として付与されている。

【0065】属性値型には、直接データの内容を値とするものと、オブジェクトが直接管理する範囲外のデータやオブジェクトへの参照情報や定義情報を値とするものがある。前者の属性値型には例えば、文字列 (STRING)、数字 (NUM) があり、後者にはデータインデクス、オブジェクトリファレンス、データリファレンス、間接データインデクス情報がある。

【0066】オブジェクトリファレンスとは、他のオブジェクトの記録場所情報であり、関係定義等で他のオブジェクトを参照する場合に用いられる。オブジェクト間に関係付けが成立するとき、属性名=ロール名が用意され、この属性の属性値として、関係先のオブジェクトリファレンスが記録される。オブジェクトとある関係を持つオブジェクトは、一般的に複数なので属性名=ロール名の属性型はリストになる。データリファレンスとは、オブジェクトが管理するデータの記録場所を示す情報である。

【0067】ここで、本実施形態において、オブジェクトの属性値として用いられ、データアクセスのために活用される、各インデクスの概念、記述方法、定義内容につき、更に以下に説明する。

【0068】ここで、インデクスとは、オブジェクトが管理するデータを細分化部品化して識別子をつけて管理するための指標である。

【0069】まず、データインデクスには、オブジェクトが管理する実際のデータにアクセスするために必要なデータの記録場所とアクセス方法に関する情報とが、属性値として記録される。データインデクスの場合の属性値の内容は、以下に示すとおりである。

【0070】{データタイプ、アクセス型、アクセス関数、引数} ここで、データタイプはデータの種類を示し、アクセス型は関数が書き込みか読み込みかを示し、アクセス関数はアクセスするための関数名が記述される。引数には、この属性に書き込みを行う場合に必要な情報を書き込む。

【0071】データインデクスを用いれば、任意のデータの内容や、その一部すなわちデータ部品を、あたかも管理オブジェクトの属性値であるかのように参照や書き込みができる。

【0072】次に、間接データインデクスとは、他のインデクスを含む演算式により定義され、間接的に、同一オブジェクト内または関係を有する他のオブジェクト内の、他の属性の属性値を参照し、これを加工して値とするインデクスデータである。尚、ここで、演算式とは、相対的なインデクス記述で他の値を参照する式をいう。

【0073】かかる間接データインデクスは、インデクス記述と演算子の組み合わせで記述される。例えば“SUM (子文章, 行数)”といった場合、関連文章の行数を総和演算することを示している。

【0074】ここで、インデクス記述とは、あるオブジェクトの特定の属性を指示する記述で、本実施形態では、属性名や関係ロール名の文字列と記号“.”を用いて、以下に示すように、インデクス記述を行う。

【0075】オブジェクト. (属性名や関係ロール名) \*\*\*. 属性名

この表記は、左から“.”を識別子として、1または複数の属性や関係ロール名をたどって、所望する属性にた



ト間に成立する関係に関する定義および実体の管理を行う関係管理手段 1 3 0 とを備えている。

【0 0 4 0】以下に、オブジェクト管理部 1 0 0 に属するテーブルに関して説明する。尚、ここにいうテーブルとは、各構成要素が管理する種々の情報や、その対応付けを示すために用いる実体の名称であって、いわゆるテーブルロードモジュールに限定されるものではなく、その論理編成は問わず、またその格納場所も、メモリ、ディスクなど、アクセスの頻度や大きさに応じて任意に決定される。

【0 0 4 1】属性定義テーブル T 1 0 は、クラス管理手段 1 1 0 が管理し、当該クラスに含まれる個々のオブジェクトの属性定義が記録されている。

【0 0 4 2】属性値記録テーブル T 2 0 は、インスタンス管理手段 1 2 0 が管理し、オブジェクトの属性値に関する情報が記録されている。

【0 0 4 3】クラス定義テーブル T 6 0 は、クラス管理手段 1 1 0 が管理し、クラスに関する属性の構成情報が記録されている。即ち、各クラスごとに、当該クラスの有する属性情報である属性名、属性型、属性値型、初期値、アクセス権などが定義されている。かかる具体的定義内容についての詳細は後述する。

【0 0 4 4】オブジェクト管理テーブル T 7 0 は、インスタンス管理手段 1 2 0 が管理し、実体であるインスタンスに関する情報が定義されている。

【0 0 4 5】関係管理テーブル T 8 0 は、関係管理手段 1 3 0 が管理し、オブジェクト間の関係定義情報が記録されている。

【0 0 4 6】尚、ここで関係とは、オブジェクト間に成立する関係を示す。ここで、オブジェクト A とオブジェクト B との関係 R は、以下のように表現される。

【0 0 4 7】A - r 1 - (関係 R) - r 2 - B

ここで、r 1、r 2 は関係の一方の終端を示すものとしてロールと呼ぶ。この場合、r 1 は、オブジェクト A から見た関係 R のロール名、r 2 は、オブジェクト B から見た関係 R のロール名であり、これらロール名を用いることによって、関係を有する相手先のオブジェクトをポイントすることが可能となる。また本来、関係自体には方向性はないが、以下の説明においては、識別のため、上記の例における A を元クラス、B を先クラス、r 1 を元ロール、r 2 を先ロールと称する。即ち、関係定義とは、かかるロールや先クラス、元クラスの組み合わせを定義するものである。

【0 0 4 8】図 3 に、関係＝編集作業の場合の関係定義の一例を示す。

【0 0 4 9】今、オブジェクト東京太郎と東京新聞との間に、編集作業という関係が成り立つとすると、元クラスに属する東京太郎にとって、“編集対象”は東京新聞であり、また、先クラスに属する東京新聞にとって、“編集者”は東京太郎となる。ここで、図 3 中の属性間制

約とは、該当オブジェクトの属性間の制約条件であり、関係がオブジェクト間に挿入されるときに、同時に挿入される。

【0 0 5 0】次に、データ登録部 2 0 0 は、オブジェクト管理部 1 0 0 により生成されたオブジェクトに対してデータの登録を行う。

【0 0 5 1】また、データアクセス部 3 0 0 は、与えられたオブジェクトの属性に対して、データ登録部 2 0 0 により登録されたデータの参照や書き込みなどの処理を行う。データアクセス手順格納テーブル T 9 0 は、データアクセス部 3 0 0 が管理し、オブジェクトに対する属性値の取得手段が記録されている。

【0 0 5 2】アクセス関数テーブル T 1 0 0 は、データアクセス部 3 0 0 が管理し、データにアクセスするための関数が、アクセス対象別に記録されている。

【0 0 5 3】インデクス管理部 4 0 0 は、本発明の中心部分を成す構成であり、所望するデータにアクセスするためのパスの記述である各種インデクスの定義、及び指定されたインデクスに対するデータ格納場所の対応付けの情報を管理する。インデクス管理部 4 0 0 は、下位構成として、インデクス管理手段 4 1 0、間接データインデクス管理手段 4 2 0、上書き手段 4 3 0、メタインデクス管理手段 4 4 0 とを有している。

【0 0 5 4】インデクス管理手段 4 1 0 は、与えられたインデクスと、オブジェクト、属性情報、および、データ格納場所との対応付けを管理し、各種インデクスを使用したデータアクセスの種類に応じ、必要な各テーブルを参照しつつ、アクセスのために必要な処理を行う。尚、以下の説明中において、[ ] 中の記載はクラス名を示し、「」中の記載はインスタンス名を示すものとする。

【0 0 5 5】格納場所管理テーブル T 9 0 は、インデクス管理手段 4 1 0 がかかる管理に用いるテーブルであり、各オブジェクト毎に属性に対応する 1 の格納場所情報を記録したテーブルである。

【0 0 5 6】ここで、格納場所情報とは、オブジェクトに対して実際にデータが格納されている場所を一意に指定する文字列である。属性名と関係ロール名を用いて左から区切り文字“#”を用いて記述する。格納場所について、値が複数ある場合は許されないの、かかる場合には、直接番号指定をする。

【0 0 5 7】例えば

“「報告書」#子文章#2”

は、

“報告書の、子文章の、2 番目の位置”

を示している。

【0 0 5 8】かかる格納場所情報からは、例えば格納場所定義に対するアクセス関数を定義する方法等により、アクセスする手順を容易に生成することが可能である。

【0 0 5 9】次に、間接データインデクス管理手段 4 2

ンデクスに従い、オブジェクト属性情報およびデータ格納場所情報を用いて、前記データ部品の検索を行うインデクス検索ステップと、与えられた記述から、前記オブジェクトの 1 または複数の前記属性の情報を含むクラス情報と、該クラスに属するインスタンス情報を用いて、対応する属性情報を出力するオブジェクト検索ステップと、前記データ部品についてのデータ格納場所情報に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセスステップとを含み、前記インデクス検索ステップは、さらに、前記インデクスに対して、複次的に付与したメタインデクスと、該メタインデクスの基礎となるインデクスとの対応関係を用いて、与えられた前記インデクス記述から、必要に応じて一次レベルのインデクスへの展開を行うメタインデクス検索ステップを含み、与えられた前記メタインデクスに基づいて、前記対応関係により示される前記オブジェクトの属性を用いて、前記データ部品へのアクセスを行うことを特徴とするものである。

【0033】上記構成によれば、アプリケーションのビューにより近く、分かりやすく曖昧なインデクスを使用したデータアクセスが可能となる。つまり、データやアプリケーションのビューの変更から独立して、柔軟にデータアクセスが行えることにより、部品レベルのデータアクセスを容易かつ効率的に行うことが可能となるのである。

【0034】更に、本発明の記憶媒体（請求項 1 1）は、格納データに対応して登録されたオブジェクトの属性を用いて、前記格納データの内部情報を、任意の格納単位に分割したデータ部品に対して、該データ部品にアクセスするためのパスを記述する、与えられたインデクスに従い、オブジェクト属性情報およびデータ格納場所情報を用いて、前記データ部品の検索を行うインデクス検索ステップと、与えられた記述から、前記オブジェクトの 1 または複数の前記属性の情報を含むクラス情報と、該クラスに属するインスタンス情報を用いて、対応する属性情報を出力するオブジェクト検索ステップと、前記データ部品についてのデータ格納場所情報に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセスステップとを含み、これらの処理をコンピュータに実行させることを特徴とするものである。かかる記憶媒体によれば、アプリケーションのビューにより近く、分かりやすく曖昧なインデクスを使用したデータアクセスが可能となる。つまり、データやアプリケーションのビューの変更から独立して、柔軟にデータアクセスが行えることにより、部品レベルのデータアクセスを容易かつ効率的に行うことが可能となるのである。

【0035】更に、本発明の記憶媒体（請求項 1 2）は、情報管理データを記録したコンピュータ読み取り可能な記録媒体であって、前記情報管理データは、少なく

とも、前記記録媒体に形成される、格納データに対応して登録されたオブジェクトの、クラス毎に定義された 1 または複数の属性情報を格納する属性定義テーブルと、前記属性名に対応して、前記格納データの内部情報を、任意の格納単位に分割したデータ部品の格納場所情報を格納する格納場所管理テーブルと、前記格納場所情報に対応して、前記属性の属性値を格納する属性値記録テーブルと、前記データ部品にアクセスするためのパスを記述するインデクスと、該インデクスに対して、さらに複次的に付与したメタインデクスとの対応関係を格納するメタインデクステーブルとに記録され、前記属性の情報は、少なくとも、前記オブジェクトの属性名、属性型、及び属性値型を含み、さらに、前記属性値型には、少なくとも、前記インデクスを示す値を含み、前記インデクスの記述には、前記属性名の文字列を含み、かかる情報管理データを用いて、コンピュータに処理を行わせることを特徴とするものである。かかる記憶媒体によれば、アプリケーションのビューにより近く、分かりやすく曖昧なインデクスを使用したデータアクセスが可能となる。つまり、データやアプリケーションのビューの変更から独立して、柔軟にデータアクセスが行えることにより、部品レベルのデータアクセスを容易かつ効率的に行うことが可能となるのである。

【0036】

【発明の実施の形態】以下、本発明の実施形態について、図面を参照しながら詳細に説明する。

【0037】図 1 に示すように、本実施形態に係る情報管理装置は、オブジェクトに関連する情報の定義、生成、管理を行うオブジェクト管理部 100 と、オブジェクトに属するデータの登録を行うデータ登録部 200 と、データベース等に保有される実際のデータ（以下、実データという。）へのアクセスを行うデータアクセス部 300 と、オブジェクト属性に対する各種インデクスの定義、管理を行うインデクス管理部 400 と、オブジェクトにおける変更に関連する他のオブジェクトに反映させる変更管理部 500 と、新しく組み込まれたクラスと既存のクラス間の対応付けを行うクラス間情報管理部 600 と、外部アプリケーションと既存オブジェクト間のインターフェースをとるプラグイン部 700 と、オブジェクトの内容の参照や更新を行う入出力部 800 とを備えている。

【0038】次に、各々の構成について、図 2 を用いて、詳細に説明する。

【0039】オブジェクト管理部 100 は、初期化処理として、クラス定義や関係定義を行い、これらの定義に基づいてインスタンスを生成し、オブジェクト間の関係の挿入を行う。そして、さらに下位構成として、クラスに関する定義、管理を行うクラス管理手段 110、実体となったオブジェクト、即ちインスタンスの生成、管理を行うインスタンス管理手段 120、複数のオブジェク



理を行う新クラス管理手段と、前記新たなクラスと、既存のクラスとの間の関連付けを記述したオブジェクト変換ルールを保持し、前記オブジェクト管理部での、既存のクラス内の前記インスタンス生成に応じて、前記変換ルールに基づき、前記新たなクラス内に対応するインスタンス生成を行うクラス間情報変換手段とを少なくとも有するクラス間情報管理部を有することにより、既存のクラスと、新しく組み込まれたクラスとの間の対応関係を保持することとなり、新たなクラスが新設された場合にも、既存のオブジェクト体系の内容を変換して利用することが可能となる。つまり、新たなクラスが生じた場合でもメンテナンスを容易にすることができる。

【0029】また、請求項4の発明においては、上記情報管理装置は、さらに、外部アプリケーションのビューと、既存のオブジェクト体系との対応に基づいて予め定義された、前記メタインデックスの情報および／又は前記クラス間情報管理部の新たなクラス定義と前記変換ルールを用いて、前記外部アプリケーションから与えられた前記インデックスの示すオブジェクト属性と、既存のオブジェクトに登録されたオブジェクト属性との間の変換を動的に行うプラグイン部を有することにより、外部アプリケーションからのビューが変更、新設されて既存のオブジェクト体系との間で、不整合がある場合でも、これらの間のギャップを埋め合わせることにより、新たなアプリケーションの組み込みを容易にすることができる。

【0030】また、請求項5の発明は、格納データに対応するオブジェクトに登録し、前記格納データの内部情報を、前記オブジェクトの属性を用いて任意の格納単位に分割したデータ部品の情報の管理を行う情報管理装置であって、前記オブジェクトの1又は複数の前記属性情報を記録する属性定義テーブルを少なくとも用いて、前記オブジェクトのクラス情報の登録及び管理を行うクラス管理手段、前記属性情報に対応する属性値を、対応するデータ部品の格納場所指定とともに記録する属性値記録テーブルを少なくとも用いて、該クラスに属するインスタンス情報の登録及び管理を行うインスタンス管理手段、および、複数の前記オブジェクト間に成立する関係の情報を記録する関係定義テーブルを少なくとも用いて、前記関係情報の登録及び管理を行う関係管理手段を少なくとも有するオブジェクト管理部と、オブジェクト属性に対応した、前記データ部品の登録を行うデータ登録部と、前記データ部品にアクセスするためのパスを記述するインデックスと、該インデックスに複次的に付与したメタインデックスとの対応関係を記録したメタインデックステーブルを少なくとも用いて、与えられたインデックス記述から、必要に応じて一次レベルのインデックスへの展開を行うメタインデックス管理手段、前記属性と、前記データ部品の格納場所指定との対応付けを記録する格納場所管理テーブルを少なくとも用いて、メタインデックス管理手段から与えられたインデックス記述から、前記データ部

品の格納場所指定により、前記データ部品に対応する前記属性値記録テーブルの属性値にアクセスするインデックス管理手段、1のオブジェクト内の他の属性、または前記関係を介する他のオブジェクトの属性から間接的に指し示された前記データ部品を用いた間接データインデックスから示されるオブジェクト属性を用いて前記データ部品へのアクセスを行い、得られた前記データ部品から、さらに、前記属性定義テーブルの前記間接データインデックスの記述に従い、仮想的なデータ部品の内容の計算を行う間接データインデックス管理手段、および与えられた前記オブジェクト属性が、書き込み可能な場合には、書き込むべき属性値を算出する上書き手段を少なくとも有するインデックス管理部と、前記データ部品についての格納場所指定に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセス部と、前記データ部品に対する変更を監視し、該変更の検知を行う変更検知手段、前記変更の検知が通知された際には、変更反映先オブジェクトへの変更内容の通知を行う変更通知手段、前記通知の内容に従い、変更反映先オブジェクトの属性値の変更を行う変更反映手段を少なくとも有する変更管理部と、新たなクラス情報の定義及び管理を行う新クラス管理手段、および前記新たなクラスと既存のクラスとの間の関連づけを記述したオブジェクト変換ルールを記録するクラス間情報変換ルールテーブルを少なくとも用いて、前記既存のクラス内の前記インスタンス生成に応じて、前記変換ルールに基づき、前記新たなクラス内に対応する新たなインスタンス生成を行うクラス間情報変換手段を少なくとも有するクラス間情報管理部と、前記メタインデックス情報および／または前記クラス間情報変換ルール情報に基づき、外部アプリケーションから与えられたインデックスの示すオブジェクト属性と、既存のオブジェクトに登録されたオブジェクト属性との間の変換を動的に行うプラグイン部と、前記オブジェクトの内容の外部からのアクセスを行う入出力手段、および前記オブジェクトの内容表示を行う表示手段を少なくとも有する入出力部とを少なくとも有することを特徴とするものである。

【0031】上記構成によれば、アプリケーションのビューにより近く、分かりやすく曖昧なインデックスを使用したデータアクセスが可能となる。つまり、データやアプリケーションのビューの変更から独立して、柔軟にデータアクセスが行えることにより、部品レベルのデータアクセスを容易かつ効率的に行うことが可能となるのである。

【0032】また、本発明方法（請求項6）は、格納データに対応するオブジェクトに登録し、前記格納データの内部情報を、前記オブジェクトの属性を用いて任意の格納単位に分割したデータ部品の情報の管理を行う情報管理装置を用いた情報検索方法であって、前記データ部品にアクセスするためのパスを記述する、与えられたイ

になされたものである。

【0017】そして、その目的とするところは、抽象レベルの柔軟なインデックスを用いることにより、データやアプリケーションのビューの変更から独立して、部品レベルのデータアクセスが容易かつ効率的に行え、データの共用性、再利用性を向上させることを可能とする情報管理装置及び情報検索方法を提供することにある。

【0018】また、他の目的は、他の属性や他のオブジェクトに属するデータ部品を関連づけたりまとめて演算を行ったりすることのできるインデックスを用いることにより、関連するデータ部品の一括したアクセス、処理を行って、データの操作時における効率の向上を図り、他のアプリケーションからのアクセスも容易にすることにある。

【0019】また、他の目的は、既存のクラスと、新しく組み込まれたクラスとの間の対応関係を保持することにより、新たなクラスが新設された場合にも、既存のオブジェクト体系の内容を変換して利用し、データの利用度の向上を図ることにある。

【0020】また、他の目的は、外部アプリケーションからのビューが変更、新設されて既存のオブジェクト体系との間で、不整合がある場合でも、これらの間のギャップを埋め合わせることにより、新たなアプリケーションの組み込みを容易にすることにある。

【0021】更に、他の目的は、抽象レベルの柔軟なインデックスを用いることにより、データやアプリケーションのビューの変更から独立して、部品レベルのデータアクセスが容易かつ効率的に行え、データの共用性、再利用性を向上させることを可能とする情報検索プログラムを格納したコンピュータ読み取り可能な記録媒体を提供することにある。

【0022】更に、他の目的は、抽象レベルの柔軟なインデックスを用いることにより、データやアプリケーションのビューの変更から独立して、部品レベルのデータアクセスが容易かつ効率的に行え、データの共用性、再利用性を向上させることを可能とする情報管理データを格納したコンピュータ読み取り可能な記録媒体を提供することにある。

【0023】

【課題を解決するための手段】要するに、本発明装置（請求項1）は、格納データに対応するオブジェクトを登録し、前記格納データの内部情報を、前記オブジェクトの属性を用いて任意の格納単位に分割したデータ部品の情報の管理を行う情報管理装置であって、前記オブジェクトの1または複数の前記属性の情報を含むクラス情報と、該クラスに属するインスタンス情報の管理を行うオブジェクト管理部と、前記データ部品にアクセスするためのパスを記述するインデックスの情報の管理を行うインデックス管理部と、前記データ部品についてのデータ格納場所情報に基づいて、前記格納データ中の前記データ

部品へのアクセスを行うデータアクセス部とを有し、前記インデックス管理部は、さらに、前記インデックスに対して、複次的に付与したメタインデックスと、該メタインデックスの基礎となるインデックスとの対応関係を保持するメタインデックス管理手段を有し、与えられた前記メタインデックスに基づいて、前記対応関係により示される前記オブジェクトの属性を用いて、前記データ部品へのアクセスを行うことを特徴とするものである。

【0024】上記構成によれば、アプリケーションのビューにより近く、分かりやすく曖昧なインデックスを使用したデータアクセスが可能となる。つまり、データやアプリケーションのビューの変更から独立して、柔軟にデータアクセスが行えることにより、部品レベルのデータアクセスを容易かつ効率的に行うことが可能となるのである。

【0025】尚、ここでいうオブジェクトとは、名前により区別される情報記録の単位である。ユーザーは、データベース等にあるデータに対応するオブジェクトを登録することにより、このオブジェクトをメタデータとして、本物のデータや、その一部分であるデータ部品にアクセスを行う。

【0026】また、ここで、クラスとは、実体であるオブジェクトのテンプレートであり、オブジェクト（インスタンス）が管理する属性情報のデフォルトの定義を有している。また、クラス定義においては、クラス間に、親クラスおよびサブクラスという形態で、階層構造を持った定義を行うことが可能である。また、上位と下位の階層の間では、階層関係に基づいて、親クラスとサブクラス間で属性を共有する、いわゆる承継が可能である。尚、ここで、オブジェクト、クラス、インスタンス等の言葉の意味内容は、オブジェクト指向技術において一般的に用いられているものと同義である。

【0027】また、請求項2の発明においては、前記オブジェクト管理部は、さらに、複数のオブジェクト間に成立する関係の情報の管理を行う関係管理手段を有し、前記インデックス管理部は、さらに、1のオブジェクト内の他の属性、又は前記関係を介する他のオブジェクトの属性から、間接的に指し示された前記データ部品を用いた間接データインデックスの情報の管理を行う間接データインデックス管理手段を有し、与えられた前記間接データインデックスより示されるオブジェクト属性を用いて、前記データ部品へのアクセスを行い、得られた前記データ部品から、さらに、前記間接データインデックスの記述に従い、新たなデータ部品を生成することにより、他の属性や他のオブジェクトに属するデータ部品を関連づけたりまとめて演算を行ったりすることとなり、関連するデータ部品の一括したアクセス、処理を行うことができる。

【0028】また、請求項3の発明においては、上記情報管理装置は、さらに、新たなクラス情報の定義及び管

10

20

30

40

50

利用を可能とするためのソフトウェア構成理念であるといえる。そして、かかるオブジェクト指向でシステムを構築することにより、外部に対しては、アプリケーションからデータに対する視点であるビューを提示することによって、アプリケーションからの独立性を高めているのである。

【0005】こうしたオブジェクト指向技術の目的とするところであるデータの共用、再利用性をさらに高めるために、保有するデータの内部データを細分化し、これを必要に応じたデータ部品として扱い、他のファイル等のデータの一部分として再利用、活用する技術に、OpenDoc (「OpenDoc genesis 〜ソフトウェア新時代の幕開け〜」、新居雅行著、ソフトバンク社発行、ISBN4-89052-823-7) や、OLE (Object Linking and Embedding, 「Understanding ActiveX and OLE」、デビッド・チャペル著、アスキー社発行、ISBN 4-7561-1706-6) がある。

【0006】以下に、OLEを例にとって、文書データを用いた場合の、データ部品の再利用の具体的操作につき説明する。尚、かかる具体的操作の態様としては、以下に示す部品オブジェクトの他のオブジェクトへの貼り付けの他、埋め込み、リンク等、すべてのデータ加工を含む。

【0007】まず、データ部品としての部品オブジェクトを、他のファイル等のオブジェクトの一部として貼り付けを行う際には、はじめに、オブジェクト定義から必要となる所定の属性を持つデータの中の、特定の属性値を有するデータ部品にまずアクセスしていた。

【0008】一般に、この場合の、かかるデータ部品へのアクセスにおいては、さらにインデックスを用いることにより、アクセスの効率化、最適化が図られていた。即ち、頻繁に行われてコストのかかるアクセスであるにも拘わらず、特定の属性値を持つオブジェクト（インスタンス）が少ないために、アクセスの的中率が低い場合には、該当するデータ部品へたどり着くまでのパス記述であるインデックスを構築し、アクセスの改善を図っていたのである。つまり、あるクラスに属するオブジェクト

（インスタンス）に包含される特定の属性について、所望する属性値を直接指し示すインデックスを、限定要素として定義しておくことにより、クラス内の全オブジェクト（インスタンス）への無駄なアクセスをすることなく、直ちに特定の属性値を有するオブジェクト、即ちデータ部品のみにアクセスすることを可能としていた。

【0009】このようにして、データ部品へのアクセスの際には、外部アプリケーションが、オブジェクトの属性、または属性値までを直接用いるか、もしくは、上述の特定のデータ部品を直接指し示すインデックスを用いてデータアクセスを行っていたのである。

【0010】そして、アクセスの結果得られたデータ部

品の操作に関しても、そのデータ部品毎にユーザーが貼り付け処理を直接指示することを要していた。従って、例えば、複数のデータ部品の間に一定の関連を有し、ファイル間での貼り付けの定型化が求められる場合であっても、個々のデータ部品毎の貼り付け操作が必要であった。

【0011】また、保有するオブジェクトの内容が変更された場合には、特定データ部品を指し示すインデックスにおけるパス記述も変更となるため、該当データをポイントするインデックスをサーチし、関連するインデックスの再定義を行っていた。

【0012】更に、外部アプリケーションのビューが変更された場合には、該当するオブジェクト（インスタンス）の再定義を行っていた。

【0013】

【発明が解決しようとする課題】このように、上述したOLE等の従来の部品データ管理および処理技術には、以下の問題点があった。即ち、第一に、従来における、データアクセスのためのインデックスは、あくまで、インスタンスレベルで、特定属性に対して付与された直接的、一次的なインデックスであったため、インスタンスの内容が変更された場合には、インデックスを定義し直さなければ、アクセスすることができず、外部アプリケーションやユーザーからのオブジェクトの変更への対応が困難であった。

【0014】また、第二に、従来のインデックスは、あくまで、インスタンスの一つの属性をポイントするものであった。このため、従来よりオブジェクト間の関係という概念はあったにも拘わらず、複数のオブジェクト間の関係情報をインデックス中に持たなかったため、データアクセスにおいては、個々のデータ部品ごとに行わなければならない、複数のデータ部品のデータ操作に定型性がある場合などに対応することができなかった。

【0015】また、第三に、部品データとしてのオブジェクトが発行されると、システム側が、OID (Object ID) として識別子を付与するが、かかる識別子は特に他のアプリケーションやユーザーがデータアクセスのために認識、使用するのには困難であり、アプリケーションのビューにこれを対応させるのは容易ではなく、データ部品の再利用を困難なものとしていた。

【0016】以上のように、本発明は、従来技術における、該当データまたはデータ部品をインスタンスレベルで直接指し示すインデックスについての情報を、別個独立に管理していたために、オブジェクトの変更やアプリケーションのビューの変更があった場合に、その対応が困難であり、また、外部から該当データをアクセスする場合にも、必要データ部品ごとに、直接アクセスを行わなければならない、データアクセスが非効率であり、さらに、他のアプリケーション等から柔軟なデータアクセスを行うのが困難であったという問題点を解決するため

記オブジェクトの 1 または複数の前記属性情報を記録する属性定義テーブルの前記間接データインデクスの記述に従い、仮想的なデータ部品の内容の計算を行う間接データインデクス検索ステップと、

与えられた前記オブジェクト属性が、書き込み可能な場合には、書き込むべき属性値を算出する上書きステップと、

与えられた記述から、少なくとも、前記属性定義テーブルを少なくとも用いて、オブジェクトのクラス情報と、該クラスに属するインスタンス情報から、対応する属性情報を出力するオブジェクト検索ステップと、

オブジェクト属性に対応した、前記データ部品の登録を行うデータ登録ステップと、

前記データ部品についてのデータ格納場所情報に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセスステップと、

前記データ部品に対する変更を監視し、該変更の検知を行う変更検知ステップと、

前記変更の検知が通知された際には、変更反映先オブジェクトへの変更内容の通知を行う変更通知ステップと、前記通知の内容に従い、変更反映先オブジェクトの属性値の変更を行う変更反映ステップと、

および新たなクラスと既存のクラスとの間の関連づけを記述したオブジェクト変換ルールを記録するクラス間情報変換ルールテーブルを少なくとも用いて、前記既存のクラス内の前記インスタンス生成に応じて、前記変換ルールに基づき、前記新たなクラス内に対応する新たなインスタンス生成を行うクラス間情報変換ステップと、前記メタインデクス情報および／または前記クラス間情報変換ルール情報に基づき、外部アプリケーションから与えられたインデクスの示すオブジェクト属性と、既存のオブジェクトに登録されたオブジェクト属性との間の変換を動的に行うプラグインステップと、

前記オブジェクトの内容の外部からのアクセスを行う入出力ステップと、

および前記オブジェクトの内容表示を行う表示ステップとを含むことを特徴とする情報検索方法。

【請求項 11】 格納データに対応して登録されたオブジェクトの属性を用いて、前記格納データの内部情報を、任意の格納単位に分割したデータ部品に対して、該データ部品にアクセスするためのパスを記述する、与えられたインデクスに従い、オブジェクト属性情報およびデータ格納場所情報を用いて、前記データ部品の検索を行うインデクス検索ステップと、

与えられた記述から、少なくとも、前記オブジェクトの 1 または複数の前記属性情報を含むクラス情報と、該クラスに属するインスタンス情報を用いて、対応する属性情報を出力するオブジェクト検索ステップと、オブジェクト属性に対応した、前記データ部品の登録を行うデータ登録ステップと、

前記データ部品についてのデータ格納場所情報に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセスステップとを含み、

これらの処理をコンピュータに実行させることを特徴とする情報検索プログラムを格納したコンピュータ読み取り可能な記録媒体。

【請求項 12】 情報管理データを記録したコンピュータ読み取り可能な記録媒体であって、前記情報管理データは、少なくとも、前記記録媒体に形成される、

格納データに対応して登録されたオブジェクトの、クラス毎に定義された 1 または複数の属性情報を格納する属性定義テーブルと、

前記属性名に対応して、前記格納データの内部情報を、任意の格納単位に分割したデータ部品の格納場所情報を格納する格納場所管理テーブルと、

前記格納場所情報に対応して、前記属性の属性値を格納する属性値記録テーブルと、

前記データ部品にアクセスするためのパスを記述するインデクスと、該インデクスに対して、さらに複次的に付与したメタインデクスとの対応関係を格納するメタインデクステーブルとに記録され、

前記属性の情報は、少なくとも、前記オブジェクトの属性名、属性型、及び属性値型を含み、

さらに、前記属性値型には、少なくとも、前記インデクスを示す値を含み、

前記インデクスの記述には、前記属性名の文字列を含むことを特徴とする情報管理データを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、情報管理装置及び情報検索方法に関し、特に、抽象レベルの柔軟なインデクスを用いることにより、データやアプリケーションのビューの変更から独立して、部品レベルのデータアクセスが容易かつ効率的に行え、データの共用性、再利用性を向上させるための技術に関する。

【0002】

【従来の技術】 近年、情報システムの進展に伴い、オブジェクト指向の技術を用いた情報管理が注目されている。

【0003】 かかる背景には、システム化対象の拡大に比例してのアプリケーション、データ等のソフトウェア資産の膨大化や、各アプリケーションの機能要求に対応して、各々が重複した情報を保有していたという情報管理の困難性という現状がある。

【0004】 オブジェクト指向とは、このように、機能要求指向でソフトウェアを構築していた従来の理念に対して、より安定的で静的なデータを中心にソフトウェアを構築することで、情報の重複を排除し、その共用、再

前記変更の検知が通知された際には、変更反映先オブジェクトへの変更内容の通知を行う変更通知手段、前記通知の内容に従い、変更反映先オブジェクトの属性値の変更を行う変更反映手段を少なくとも有する変更管理部と、新たなクラス情報の定義及び管理を行う新クラス管理手段、

および前記新たなクラスと既存のクラスとの間の関連付けを記述したオブジェクト変換ルールを記録するクラス間情報変換ルールテーブルを少なくとも用いて、前記既存のクラス内の前記インスタンス生成に応じて、前記変換ルールに基づき、前記新たなクラス内に対応する新たなインスタンス生成を行うクラス間情報変換手段を少なくとも有するクラス間情報管理部と、

前記メタインデックス情報および／または前記クラス間情報変換ルール情報に基づき、外部アプリケーションから与えられたインデックスの示すオブジェクト属性と、既存のオブジェクトに登録されたオブジェクト属性との間の変換を動的に行うプラグイン部と、

前記オブジェクトの内容の外部からのアクセスを行う入出力手段、

および前記オブジェクトの内容表示を行う表示手段を少なくとも有する入出力部とを少なくとも有することを特徴とする情報管理装置。

【請求項 6】 格納データに対応するオブジェクトを登録し、前記格納データの内部情報を、前記オブジェクトの属性を用いて任意の格納単位に分割したデータ部品の情報の管理を行う情報管理装置を用いた情報検索方法であって、

前記データ部品にアクセスするためのパスを記述する、与えられたインデックスに従い、オブジェクト属性情報およびデータ格納場所情報を用いて、前記データ部品の検索を行うインデックス検索ステップと、

与えられた記述から、前記オブジェクトの 1 または複数の前記属性の情報を含むクラス情報と、該クラスに属するインスタンス情報を用いて、対応する属性情報を出力するオブジェクト検索ステップと、

前記データ部品についてのデータ格納場所情報に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセスステップとを含み、

前記インデックス検索ステップは、さらに、前記インデックスに対して、複次的に付与したメタインデックスと、該メタインデックスの基礎となるインデックスとの対応関係を用いて、与えられた前記インデックス記述から、必要に応じて一次レベルのインデックスへの展開を行うメタインデックス検索ステップを含み、

与えられた前記メタインデックスに基づいて、前記対応関係により示される前記オブジェクトの属性を用いて、前記データ部品へのアクセスを行うことを特徴とする情報検索方法。

【請求項 7】 前記インデックス検索ステップは、さらに、1 のオブジェクト内の他の属性、又は、複数のオブジェクト間に成立する関係を介する他のオブジェクトの属性から、間接的に指し示された前記データ部品を用いた間接データインデックスより示されるオブジェクト属性を用いて、前記データ部品へのアクセスを行い、得られた前記データ部品から、さらに、前記間接データインデックスの記述に従い、新たなデータ部品を生成する間接データインデックス検索ステップを含むことを特徴とする請求項 6 記載の情報検索方法。

【請求項 8】 上記情報検索方法は、さらに、新たなクラスが定義された際には、前記新たなクラスと、既存のクラスとの間の関連付けを予め記述したオブジェクト変換ルールに基づいて、既存のクラス内の前記インスタンス生成に応じて、前記新たなクラス内に対応するインスタンス生成を行うクラス間情報変換ステップを含むことを特徴とする請求項 6 または 7 のいずれか記載の情報検索方法。

【請求項 9】 上記情報検索方法は、さらに、外部アプリケーションからの検索情報の入力となされた際には、外部アプリケーションのビューと、既存のオブジェクト体系との対応に基づいて予め定義された、前記メタインデックスの情報および／又は前記新たなクラス定義と前記変換ルールを用いて、前記外部アプリケーションから与えられた前記インデックスの示すオブジェクト属性と、既存のオブジェクトに登録されたオブジェクト属性との間の変換を動的に行うプラグインステップを含むことを特徴とする請求項 8 記載の情報検索方法。

【請求項 10】 格納データに対応するオブジェクトを登録し、前記格納データの内部情報を、前記オブジェクトの属性を用いて任意の格納単位に分割したデータ部品の情報の管理を行う情報管理装置を用いた情報検索方法であって、

前記データ部品にアクセスするためのパスを記述する、与えられたインデックスに従い、オブジェクト属性情報と、前記データ部品の格納場所指定との対応付けを記録する格納場所管理テーブルを少なくとも用いて、前記データ部品の検索を行うインデックス検索ステップと、

前記インデックスに対して、複次的に付与したメタインデックスと、該メタインデックスの基礎となるインデックスとの対応関係を記録するメタインデックステーブルを少なくとも用いて、与えられた前記インデックス記述から、必要に応じて一次レベルのインデックスへの展開を行うメタインデックス検索ステップと、

1 のオブジェクト内の他の属性、または前記関係を介する他のオブジェクトの属性から間接的に指し示された前記データ部品を用いた間接データインデックスから示されるオブジェクト属性を用いて前記データ部品へのアクセスを行い、得られた前記データ部品から、さらに、前

## 【特許請求の範囲】

【請求項 1】 格納データに対応するオブジェクトを登録し、前記格納データの内部情報を、前記オブジェクトの属性を用いて任意の格納単位に分割したデータ部品の情報の管理を行う情報管理装置であって、前記オブジェクトの 1 または複数の前記属性の情報を含むクラス情報と、該クラスに属するインスタンス情報の管理を行うオブジェクト管理部と、前記データ部品にアクセスするためのパスを記述するインデックスの情報の管理を行うインデックス管理部と、前記データ部品についてのデータ格納場所情報に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセス部とを有し、前記インデックス管理部は、さらに、前記インデックスに対して、複次的に付与したメタインデックスと、該メタインデックスの基礎となるインデックスとの対応関係を保持するメタインデックス管理手段を有し、与えられた前記メタインデックスに基づいて、前記対応関係により示される前記オブジェクトの属性を用いて、前記データ部品へのアクセスを行うことを特徴とする情報管理装置。

【請求項 2】 前記オブジェクト管理部は、さらに、複数のオブジェクト間に成立する関係の情報の管理を行う関係管理手段を有し、前記インデックス管理部は、さらに、1 のオブジェクト内の他の属性、又は前記関係を介する他のオブジェクトの属性から、間接的に指し示された前記データ部品を用いた間接データインデックスの情報の管理を行う間接データインデックス管理手段を有し、与えられた前記間接データインデックスより示されるオブ

ジェクト属性を用いて、前記データ部品へのアクセスを行い、得られた前記データ部品から、さらに、前記間接データインデックスの記述に従い、新たなデータ部品を生成することを特徴とする請求項 1 記載の情報管理装置。

【請求項 3】 上記情報管理装置は、さらに、新たなクラス情報の定義及び管理を行う新クラス管理手段と、前記新たなクラスと、既存のクラスとの間の関連付けを記述したオブジェクト変換ルールを保持し、前記オブジェクト管理部での、既存のクラス内の前記インスタンス生成に応じて、前記変換ルールに基づき、前記新たなクラス内に対応するインスタンス生成を行うクラス間情報変換手段とを少なくとも有するクラス間情報管理部を有することを特徴とする請求項 1 または 2 のいずれか記載の情報管理装置。

【請求項 4】 上記情報管理装置は、さらに、外部アプリケーションのビューと、既存のオブジェクト体系との対応に基づいて予め定義された、前記メタインデックスの情報および／又は前記クラス間情報管理部の新たなクラス定義と前記変換ルールを用いて、前記外部アプリケーションから与えられた前記インデ

スの示すオブジェクト属性と、既存のオブジェクトに登録されたオブジェクト属性との間の変換を動的に行うプラグイン部を有することを特徴とする請求項 3 記載の情報管理装置。

【請求項 5】 格納データに対応するオブジェクトを登録し、前記格納データの内部情報を、前記オブジェクトの属性を用いて任意の格納単位に分割したデータ部品の情報の管理を行う情報管理装置であって、前記オブジェクトの 1 又は複数の前記属性情報を記録する属性定義テーブルを少なくとも用いて、前記オブジェクトのクラス情報の登録及び管理を行うクラス管理手段、

前記属性情報に対応する属性値を、対応するデータ部品の格納場所指定とともに記録する属性値記録テーブルを少なくとも用いて、該クラスに属するインスタンス情報の登録及び管理を行うインスタンス管理手段、および、複数の前記オブジェクト間に成立する関係の情報を記録する関係定義テーブルを少なくとも用いて、前記関係情報の登録及び管理を行う関係管理手段を少なくとも有するオブジェクト管理部と、オブジェクト属性に対応した、前記データ部品の登録を行うデータ登録部と、

前記データ部品にアクセスするためのパスを記述するインデックスと、該インデックスに複次的に付与したメタインデックスとの対応関係を記録したメタインデックステーブルを少なくとも用いて、与えられたインデックス記述から、必要に応じて一次レベルのインデックスへの展開を行うメタインデックス管理手段、前記属性と、前記データ部品の格納場所指定との対応付けを記録する格納場所管理テーブルを少なくとも用いて、メタインデックス管理手段から与えられたインデックス記述から、前記データ部品の格納場所指定により、前記データ部品に対応する前記属性値記録テーブルの属性値にアクセスするインデックス管理手段、

1 のオブジェクト内の他の属性、または前記関係を介する他のオブジェクトの属性から間接的に指し示された前記データ部品を用いた間接データインデックスから示されるオブジェクト属性を用いて前記データ部品へのアクセスを行い、得られた前記データ部品から、さらに、前記属性定義テーブルの前記間接データインデックスの記述に従い、仮想的なデータ部品の内容の計算を行う間接データインデックス管理手段、

および与えられた前記オブジェクト属性が、書き込み可能な場合には、書き込むべき属性値を算出する上書き手段を少なくとも有するインデックス管理部と、

前記データ部品についての格納場所指定に基づいて、前記格納データ中の前記データ部品へのアクセスを行うデータアクセス部と、

前記データ部品に対する変更を監視し、該変更の検知を行う変更検知手段、



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-143895

(43) 公開日 平成11年(1999) 5月28日

(51) Int.Cl.<sup>6</sup>

識別記号

F I

G 0 6 F 17/30  
12/00

5 4 7

G 0 6 F 15/40  
12/00  
15/401

3 8 0 D  
5 4 7 A  
3 2 0 B

審査請求 未請求 請求項の数12 O L (全 30 頁)

(21) 出願番号

特願平9-307460

(22) 出願日

平成9年(1997)11月10日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 岩政 幹人

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(74) 代理人 弁理士 三好 秀和 (外 3 名)

(54) 【発明の名称】 情報管理装置及び情報検索方法

(57) 【要約】

【課題】 抽象レベルの柔軟なインデックスを用いることにより、データやアプリケーションのビューの変更から独立して、部品レベルのデータアクセスが容易かつ効率的に行え、データの共用性、再利用性を向上させることにある。

【解決手段】 オブジェクトの管理を行うオブジェクト管理部100と、データ部品にアクセスするインデックスの管理を行うインデックス管理部400と、データ部品へのアクセスを行うデータアクセス部300とを有し、インデックス管理部400は、インデックスに対して、複次的に付与したメタインデックスと、基礎となるインデックスとの対応関係を保持するメタインデックス管理手段440を有し、与えられたメタインデックスから、データ部品へのアクセスを行う。

